UNIVERSITÉ DE MONS
Faculté polytechnique
Département électricité

UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique
et de génie informatique

INTERACTION INTERMODALE DANS LES RÉSEAUX NEURONAUX PROFONDS POUR LA CLASSIFICATION ET LA LOCALISATION D'ÉVÉNEMENTS AUDIOVISUELS

Cross-Modal Interaction in Deep Neural Networks for Audio-Visual Event Classification and Localization

Thèse de doctorat

Mathilde BROUSMICHE

Superviseur: Dr. Stéphane Dupont (Université de Mons)
Superviseur: Prof. Jean Rouat (Université of Sherbrooke)

Janvier 2021

This thesis was supported by the European Regional Development Fund (ERDF) and CHISTERA Interactive Grounded Language Understanding (IGLU) project

Jury Members

Prof. **Xavier SIEBERT** ..... President, University of Mons
Dr. **Stéphane DUPONT** .....Supervisor, University of Mons
Prof. **Jean ROUAT** ..... Supervisor, Université de Sherbrooke
Prof. **Thierry DUTOIT** .....University of Mons
Prof. **François FERLAND** .....Université de Sherbrooke
Prof. **Giampiero SALVI** .....
. .....Norwegian University of Science and Technology

Résumé

La compréhension automatique du monde environnant a de nombreuses applications telles que la surveillance et sécurité, l'interaction Homme-Machine, la robotique, les soins de santé, etc. Plus précisément, la compréhension peut s'exprimer par le biais de différentes tâches telles que la classification et localisation dans l'espace d'évènements. Les êtres vivants exploitent un maximum de l'information disponible pour comprendre ce qui les entoure. En s'inspirant du comportement des êtres vivants, les réseaux de neurones artificiels devraient également utiliser conjointement plusieurs modalités, par exemple, la vision et l'audition.

Premièrement, les modèles de classification et localisation, basés sur l'information audio-visuelle, doivent être évalués de façon objective. Nous avons donc enregistré une nouvelle base de données pour compléter les bases actuellement disponibles. Comme aucun modèle audio-visuel de classification et localisation n'existe, seule la partie sonore de la base est évaluée avec un modèle de la littérature.

Deuxièmement, nous nous concentrons sur le coeur de la thèse: comment utiliser conjointement de l'information visuelle et sonore pour résoudre une tâche spécifique, la reconnaissance d'évènements. Le cerveau n'est pas constitué d'une "simple" fusion mais comprend de multiples interactions entre les deux modalités. Il y a un couplage important entre le traitement de l'information visuelle et sonore. Les réseaux de neurones offrent la possibilité de créer des interactions entre les modalités en plus de la fusion. Dans cette thèse, nous explorons plusieurs stratégies pour fusionner les modalités visuelles et sonores et pour créer des interactions entre les modalités. Ces techniques ont les meilleures performances en comparaison aux architectures de l'état de l'art au moment de la publication. Ces techniques montrent l'utilité de la fusion audio-visuelle mais surtout l'importance des interactions entre les modalités.

Pour conclure la thèse, nous proposons un réseau de référence pour la classification et localisation d'évènements audio-visuels. Ce réseau a été testé avec la nouvelle base de données. Les modèles précédents de classification sont modifiés pour prendre en compte la localisation dans l'espace en plus de la classification.

Mots clés: Fusion Audio-visuelle, Conditionnement de modalités, Apprentissage profond multimodale, Reconnaissance d'évènements, Localisation d'évènements

Abstract

The automatic understanding of the surrounding world has a wide range of applications, including surveillance, human-computer interaction, robotics, health care, etc. The understanding can be expressed in several ways such as event classification and its localization in space. Living beings exploit a maximum of the available information to understand the surrounding world. Artificial neural networks should build on this behavior and jointly use several modalities such as vision and hearing.

First, audio-visual networks for classification and localization must be evaluated objectively. We recorded a new audio-visual dataset to fill a gap in the current available datasets. We were not able to find audio-visual models for classification and localization. Only the dataset audio part is evaluated with a state-of-the-art model.

Secondly, we focus on the main challenge of the thesis: How to jointly use visual and audio information to solve a specific task, event recognition. The brain does not comprise a simple fusion but has multiple interactions between the two modalities to create a strong coupling between them. The neural networks offer the possibility to create interactions between the two modalities in addition to the fusion. We explore several strategies to fuse the audio and visual modalities and to create interactions between modalities. These techniques have the best performance compared to the state-of-the-art architectures at the time of publishing. They show the usefulness of audio-visual fusion but above all the contribution of the interaction between modalities.

To conclude, we propose a benchmark for audio-visual classification and localization on the new dataset. Previous models for the audio-visual classification are modified to address the localization in addition to the classification.

Keywords: Audio-visual fusion, Modality conditioning, Multimodal deep learning, Event recognition, Event localization

"What we know is a drop, what we don't know is an ocean."
Isaac Newton

Acknowledgements

These years of thesis have been rich in emotions, sharing and learning. I had the opportunity to enrich my knowledge and skills both professionally and personally. During my co-supervision, I had the chance to meet amazing people, both in Belgium and in Canada. They gave me their precious advice and I wish to express my sincere thanks to you.

First of all, I would like to thank my supervisors Stéphane Dupont and Jean Rouat who allowed me to live this experience. I thank you for your advice, your trust, your support and the time you devoted to me. You were able to be present while leaving me a great freedom of work.

I would also like to thank the members of the DPR jury, of the *comité d'accompagnement* and of my thesis jury for their advice, remarks and suggestions on my work which helped me at the different stages of my thesis.

A big thank you to Thierry Dutoit and all my colleagues from Mons and Sherbrooke for this so pleasant atmosphere in the laboratories but also for our "social" activities, your help without counting and more particularly to the colleagues from Mons for these crazy discussions during lunch time. A special thanks to Nathalie without whom we would be totally lost.

A huge thank you to the many people who reviewed this thesis. Your comments and advice helped to make this document better.

I also thank my friends for your support, encouragement, friendship and good mood. I would especially like to thank Manon who is always there to encourage me in my times of doubt and celebrate my successes with me. Thank you

again for those record time skypes when I was feeling nostalgic on the other side of the ocean.

Finalement, je ne serai pas qui je suis sans l'amour et le soutien inconditionnel de ma famille et plus particulièrement de mes parents et de mes frères. Je ne vous remercierai jamais assez pour votre présence, votre aide et vos encouragements dans les bons mais surtout dans les moments de doute. Merci Mushu pour les calinoux et les bisous guérisseurs, désolée, je n'ai pas construit BB-8 mais un jour peut-être ...

Contents

List of acronyms

Adam	Adaptive Moment Estimation	17
ANN	Artificial Neural Network	12
CN	Conditional Normalization	90
CNN	Convolutional Neural Network	6
CRNN	Convolutional Recurrent Neural Network	64
CV	Computer Vision	19
DL	Deep Learning	20
DMR	Dual Multimodal Residual	38
DNN	Deep Neural Network	3
DOA	Direction of Arrival	66
ER	Error Rate	72
FC	Fully Connected	14
FFT	Fast Fourier Transform	xv
FiLM	Feature-wise Linear Modulation	94
FNN	Feedforward Neural Network	12
GCC	Generalized Cross Correlation	65
GRU	Gated Recurrent Unit	12
IR	Impulse Response	45
LSTM	Long Short-Term Memory	8

Introduction

Living beings understand a scene through various senses such as sight, hearing, touch, etc. [1] These modalities can be both complementary and redundant. For example, the perception of speech and the lip movement are redundant and complementary as the Mcgurk effect [2] proves. It highlights the interaction between hearing and vision in speech perception and proves that speech perception is influenced by the lip movement of the speaker. The redundancy between the modalities allows better robustness, for example, in the speech perception, but also more generally in the scene understanding.

The brain has evolved to learn and operate optimally in the presence of several modalities [3]. Indeed, human beings recognize and localize objects more easily when several modalities, such as sight and hearing, are present [4]. Machine learning protocols with multimodal data would be closer to reality than unimodal protocols.

Numerous multisensory convergence zones have been identified in the brain [5]. These multisensory zones are regions where biological neurons receive inputs coming from different senses and combine them according to various temporal, spatial or even semantic constraints. Some brain regions, that were previously considered specific to the visual modality, exhibit multisensory modulation [6]. Such interference may occur even at the beginning of information processing, for example, in primary visual cortex, previously considered strictly modality-specific. Currently, neuroscience research shows that multisensory integration operates at different levels in the brain: in subcortical structures, in higher-level associative cortices and even in early cortical areas [7].

On the other hand, the artificial neuron was created in the 50s. However, results that appeal to the scientific community and the gain in popularity of Deep Neural Networks (DNNs) only occurred about ten years ago. Nowadays,

DNNs are the state of the art in many problems such as computer vision, speech and natural language processing, etc.

The number of DNN techniques applied to multimodal data increases every year (Figure 0.1). The motivation to use multimodal data is that complementary information could be extracted from each modality. This yields a richer representation and improves performance compared to using a single modality.

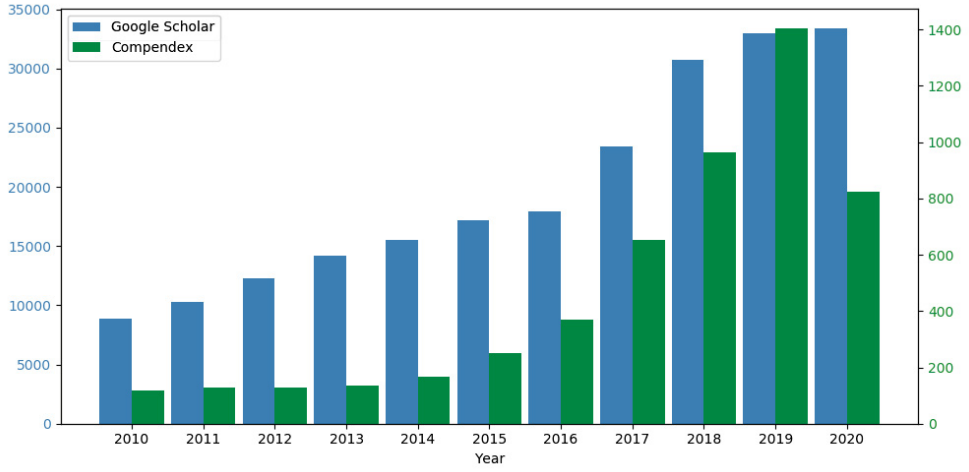


Figure 0.1. The number of publications with the words "multimodal" and "neural network" over the years with two literature databases: Google Scholar and Compendex.

The scene understanding problem can be approached with different tasks, for example, the event classification, detection and localization. The event classification consists to estimate the class, also called the category, present in a video. The event class can be very varied: someone speaking, whistling, a baby crying, someone playing a musical instrument, a car passing, a dog barking, etc. To make this task more complex, event detection was proposed. In addition to the event class, it is possible to estimate the beginning and end of the event in time. Finally, the event is localized in space, x, y coordinates or an azimuth angle are associated with each event. Depending on the task performed and the modality used, there are different problems in the literature:

video recognition, visual event detection, sound event detection, sound source localization, audio-visual event detection, etc.

Even though a lot of research uses visual or sound information, for event classification, detection and localization, the use of multimodal data was still rare until recent years. As living beings use a maximum of the available information to understand an event, DNNs should take as input multimodal data to accomplish these different tasks. There is no one way to effectively fuse information from different modalities. Inspired by the perceptual principles of the brain, multimodal models should not only be composed of a fusion but of several interactions between the visual and audio paths.

Contributions of this thesis

The original contributions of this thesis are listed below:

- A novel audio-visual event dataset was recorded to fill a gap in currently available datasets to evaluate models for the audio-visual event classification and localization tasks. The dataset is available online in the form of two folders: SECL-UMONS (recordings with a microphone array) and AVECL-UMONS (recordings with four cameras). The development of a registration procedure allowed the automation of the annotation step to facilitate the dataset creation;
- To the knowledge of the author, no audio-visual neural network for classification and localization exists. Therefore, we evaluated the audio recordings (SECL-UMONS) with a network based on audio modality only. We also slightly modified the baseline model to create a more real-time architecture;
- The joint use of audio-visual modalities is a complex challenge. Several approaches were explored. They can be grouped into 2 categories: modality fusion and modality interaction, also called modality conditioning. For the fusion:
  - We studied different state-of-the-art fusion techniques;
  - We implemented a novel technique based on attention. This fusion focuses on a modality rather than the other at every moment of a video. For example, in a scene composed of a train that whistles

in the distance, we first focus on the audio modality because the train is too small to be identified. The visual modality is then more accurate when the train is closer.

- The living being's brain includes numerous connections between the process of visual and audio information. This creates a strong coupling between the modalities. Therefore, we investigated the modality interaction in addition to the fusion. We implemented 3 different strategies:
  - First, the connection between the 2 modalities is performed at the feature level. Audio modality highlights patterns, shape, edge, color or features of the visual modalities and vice versa. Indeed, feature maps of a Convolutional Neural Network (CNN) can detect several features such as circles, lines, colors, etc. Deep feature maps can detect more complex shapes such as bicycles, cars, dogs, etc. The goal is, therefore, to give greater importance to feature maps associated with the dog, if we hear barking or to the 'car' feature map if we hear an engine sound.
  - The second proposition is implemented at the temporal level. Each time step of the audio modality interacts with each time step of the visual modality. This technique finds correlations between modalities across time. Indeed, relevant visual and audio information is not necessarily in the video at the same time. Finding related information in both modalities helps to improve classification.
  - Finally, the last technique works also at the temporal level. It models long-term dependencies. An event includes a temporal aspect and can take place over several seconds. It is necessary to take into account the temporal context at every moment. This context can be modality-specific but also audio-visual.

  The fusion based on attention and the feature-level modality conditioning are used jointly to form a first network for audio-visual event recognition, named Multi-level Attention Fusion network (MAFnet). A second model composed of the two modality interactions in the temporal domain is proposed for audio-visual event classification and detection.

- Finally, we used the results of the previous contributions to create final networks. These networks are benchmarks for the classification and localization of audio-visual events.

The papers published during the thesis are listed in Appendix A.

Organization of this dissertation

The dissertation is divided into four parts. The organization of this document is as follows:

Part 1 presents the theoretical background. First, the different deep learning concepts necessary to understand the dissertation are introduced (Chapter 1). A reader who is already familiar with the notions of deep learning is invited to go directly to Chapter 2. Then, several strategies to simultaneously exploit audio and visual data are presented (Chapter 2). This part is a general introduction to the use of audio-visual data in the literature. A deeper and detailed presentation is made for each part of the dissertation.

Part 2 presents the different event classification and localization datasets from the literature (Chapter 3) and the new dataset recorded during the thesis (Chapter 4).

Part 3 presents the Sound Event Localization and Detection (SELD) task. A detailed review of the literature for the two separate subtasks (Sound Event Detection (SED) and Sound Source Localization (SSL)) and the global task (Sound Event Localization and Detection (SELD)) is made in Chapter 5. The sound part of the new dataset is evaluated with the baseline model of the DCASE Challenge (Chapter 6).

Part 4 focuses on the fusion of audio-visual data. First, we study different state-of-the-art fusion strategies and introduce the modality conditioning and its efficiency (Chapter 8). Then, we detail two networks proposed during this thesis. On one hand, a multi-level attention network for audio-visual event classification (composed of conditioning layer and fusion based on attention) (Chapter 9). On the other hand, a network for audio-visual event detection and classification (composed of intra and inter-modality interactions and multimodal Long Short-Term Memory (LSTM))(Chapter 10). We conclude this part with final networks, built based on previous experiments, for audio-visual event classification and localization (Chapter 11).

Part I

Theoretical background

Chapter 1

Deep Neural Network

Contents

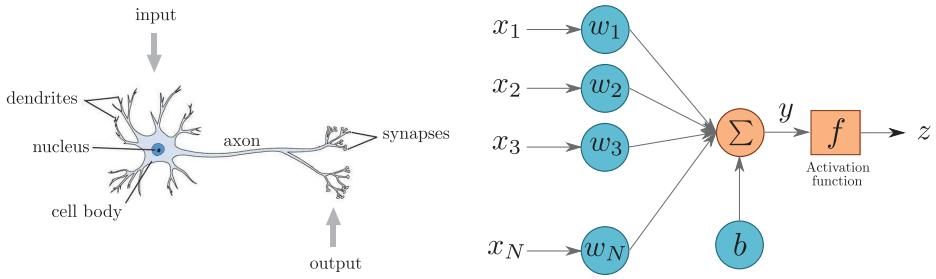
Currently, Deep Neural Networks (DNNs), based on Artificial Neural Networks (ANNs), are popular and effective to solve classification and regression tasks. They are applied to many fields, including computer vision [8–13], speech recognition [14, 15], natural language processing [16], machine translation [17, 18], medical image analysis [19–21], speech synthesis [22, 23], etc. Despite the fact that ANNs were inspired by information processing and distributed communication nodes in biological systems, the functioning is very different from the biological neurons. Specifically, ANNs tend to be static and symbolic, while the biological brain is dynamic (plastic) and analog.

In this chapter, we briefly present two types of Deep Neural Network (DNN) architectures: Feedforward Neural Network (FNN) and Recurrent Neural Network (RNN). The FNN is composed of feedforward connections, meaning the information flows from the input to the output. On the other hand, the RNN has a recursive structure, where the information can loop. In the different sections, we introduce different FNNs from the simplest to more complex structures: the perceptron, the Multilayer Perceptron (MLP) and the CNN. Then, we present two RNNs: The LSTM and the Gated Recurrent Unit (GRU). Finally, we describe a sub-network class called Attention Network as well as some normalization techniques in Deep Neural Network (DNN) architectures. This chapter is a brief reminder of the DNN basic principles, the reader can refer to different books for a more in-depth understanding [24].

1.1 Perceptron

The artificial neuron, also called perceptron [21], is a simplified model of a biological neuron. Indeed, the biological neural receives information along several dendrites, processes this information in the cell and sends an output signal along the axon (Figure 1.1a). Inspired by this mechanism, the perceptron receives several inputs modulated by weights. It generates an output by summing the modulated inputs and applying an activation function (Figure 1.1b).

Formally, given an input x of size N, the perceptron computes the output z as follows:



(a) Biological neuron **(b)** Artificial neuron

Figure 1.1. Model of a biological neural (a) and an artificial neuron, also called perceptron (b).

$$y = \left(\sum_{i=1}^{N} w_i x_i \right) + b \tag{1.1}$$

$$z = f(y) \tag{1.2}$$

where w_i are trainable weights, b is the bias and $f(.)$ is the activation function. In the initial formulation of the perceptron, the activation function was a Heaviside step function. But over the year, the concept has been generalized, and different activation functions have been proposed, for example, the Rectified Linear Unit (ReLU) function [25], the Hyperbolic Tangent (tanh) [26], the Sigmoid function [27], etc. (Figure 1.2). The activation function determines whether the neuron should be activated ("fired") or not. It can introduce non-linearity into the network and normalize the output of each neuron to a range between 1 and 0 or between -1 and 1.

1.2 Multilayer Perceptron

The Multilayer Perceptron (MLP) [28] is a mathematical function mapping some set of input values to output values. It is capable of representing highly complex functions. The MLP consists of several layers of perceptrons (Figure

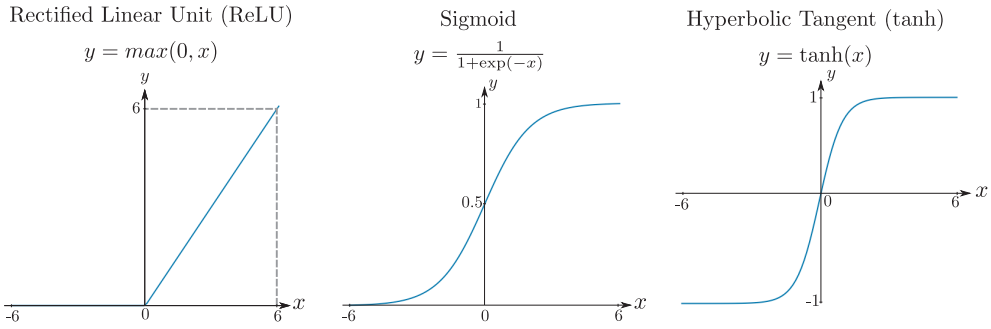


Figure 1.2. Examples of activation functions.

1.3). The first and last layers are called the input and output layers, respectively. The layers between are called hidden layers. Each layer comprises several neurons, also called units. MLPs are fully connected structures, meaning each node in one layer is connected to every node in the following layer with a weight w_{ij}. In the rest of the manuscript, these layers are interchangeably called Fully Connected (FC) or dense layer.

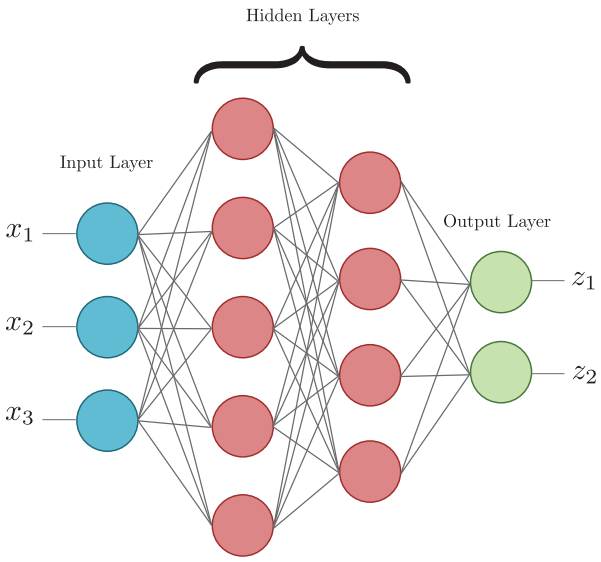


Figure 1.3. Model of MLP with 2 hidden layers.

Formally, the output of the neuron j of the l^{th} layer is expressed as:

$$y_j^l = \left(\sum_{i=1}^{N^{l-1}} w_{ij}^l h_i^{l-1} \right) + b_j^l \tag{1.3}$$

$$h_j^l = f^l \left(y_j^l \right) \tag{1.4}$$

where N^l is the total number of neurons in the layer l, w_{ij}^l is the weight applied to the feedforward connection from the neuron i of layer $l-1$ to the neuron j of layer l, b_j^l is the bias of the neuron j of the l^{th} layer and $f^l(.)$ is the activation function of the l^{th} layer.

The information is propagated from the initial input features to the output layer. The activation function of the output layer depends on the task to solve. For example, in the case of the multiclass classification (one class among several classes is chosen for each input), the Softmax non-linear function [29] is used to normalize the output of a network to a probability distribution over estimated output classes:

$$\hat{y}_j = \text{softmax}(z_j) = \frac{\exp(z_j)}{\sum_{i=1}^{N} \exp(z_i)} \tag{1.5}$$

In the case of the multilabel classification (multiple classes may be chosen for each input), as the outputs are not mutually exclusive, the Sigmoid function is used to normalize the output of the network between 0 and 1.

1.2.1 Loss function

MLPs are composed of weights, modified to estimate the correct output for a given input. To evaluate the accuracy of the output ($\hat{y}$), the output is compared to an expected result called the target (y). The error between the output and the target is computed with a loss function (C). Several loss functions are proposed in the literature depending on the task to solve.

For the multiclass classification, the categorical cross-entropy loss is computed by summing the cross-entropy for each class:

$$C = -\sum_{i=1}^{K} y_i \log(\hat{y}_i) \tag{1.6}$$

The binary cross-entropy is a particular case of the previous loss where $K = 2$. It is used for the binary classification:

$$C = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \tag{1.7}$$

In the case of the multilabel classification task, the loss is the sum of the binary cross-entropy for each class:

$$C = -\sum_{i=1}^{K} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \tag{1.8}$$

In the case of the regression task, the most used loss function is the Mean Squared Error (MSE) between the estimated output and the target:

$$C = \frac{1}{2}(\hat{y} - y)^2 \tag{1.9}$$

1.2.2 Gradient Descent

In supervised learning, the output of the MLP has to be equal to a known target. Therefore, the loss has to be as small as possible. The loss function C depends on two variables: the input X and the weights W of the MLP. As the inputs are fixed, the weights have to be modified to minimize the loss. This minimization problem can be solved with an iterative optimization algorithm, the gradient descent [30, 31]. The weights W of the network are iteratively updated with the partial derivative of the loss C with respect to the weights W:

$$W^{n+1} = W^n - \mu \left(\frac{\nabla C(X, W)}{\nabla W} \right) \tag{1.10}$$

where μ is the learning rate. It should be large enough to reach a minimum without too many iterations, but not too large to avoid oscillating around the minimum.

The gradient of C with respect to W, $\frac{\nabla C(X,W)}{\nabla W}$, is determined with the chain rule [32]. The error is backpropagated from upper layer to lower layer.

If the loss and the weight update are computed with all examples, the time to do a single gradient step becomes extremely long. On the other hand, if the weights are updated for each example, the descent is very fuzzy. The Stochastic Gradient Descent (SGD) [33] is a compromise between not enough and too many examples to compute the gradient. A small amount of data, called a batch, is used to estimate the gradient and update the weights.

Many improvements in the SGD algorithm have been proposed such as Adaptive Moment Estimation (Adam) algorithm [34]. The method computes individual adaptive learning rates for different parameters by estimating first and second moments of the gradients.

The moving average for the first and second moments and the bias correction are computed:

$$m^{n+1} = \beta_1 m^n + (1 - \beta_1) \frac{\nabla C(X, W)}{\nabla W} \tag{1.11}$$

$$v^{n+1} = \beta_2 v^n + (1 - \beta_2) \left(\frac{\nabla C(X, W)}{\nabla W} \right)^2 \tag{1.12}$$

$$\hat{m} = \frac{m^{n+1}}{1 - \beta_1} \tag{1.13}$$

$$\hat{v} = \frac{v^{n+1}}{1 - \beta_2} \tag{1.14}$$

where β_1 and β_2 are the forgetting factors.

The weights are then updated as follows:

$$w^{n+1} = w^n - \mu \frac{\hat{m}}{\sqrt{\hat{v}} + \epsilon} \tag{1.15}$$

where μ is the learning rate and ϵ a small scalar used to prevent division by zero.

1.2.3 Multi-task learning

Up to now, the network is composed of a single output layer and trained to solved one task. However, it is possible to address multiple related tasks at the same time. The neural network is train to solve several problems at once instead of having several separate neural networks. This can result in improved learning efficiency and estimation accuracy, when compared to training the models separately.

In practice, the model is composed of several output layers, one for each task (Figure 1.4). The models learns different tasks in parallel while using shared representation. The global loss is computed by taking the weighted sum of the loss of each task:

$$C = \lambda_1 C_1 + \cdots + \lambda_i C_i + \cdots + \lambda_K C_K \tag{1.16}$$

where λ_i is the weight associated to the task C_i.

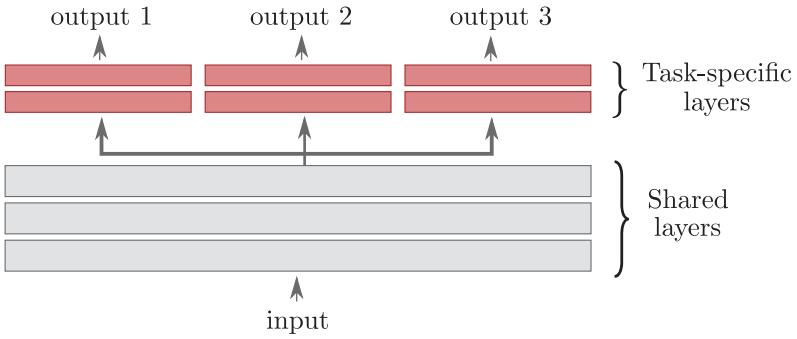


Figure 1.4. Multi-task learning network for 3 tasks.

1.3 Convolutional Neural Networks

In 1998, LeCun proposed a new kind of neural network architecture for handwritten characters recognition: the Convolutional Neural Network (CNN) [35]. The CNNs are also Feedforward Neural Networks (FNNs) but with a more complex internal architecture than the Multilayer Perceptron (MLP). They are specialized for processing data with a grid-like topology, for example, time-series data (1-D grid taking samples at regular time intervals), and image data (2-D grid of pixels). They can be interpreted as banks of Finite Impulse Response (FIR) filters implemented with neurons. The CNNs have shown enormous potential in Computer Vision (CV) tasks.

A typical CNN is composed of three types of layers: convolutional layers, pooling layers and Fully Connected (FC) layers, when used for classification tasks (Figure 1.5). The convolutional and pooling layers are explained in the following sections.

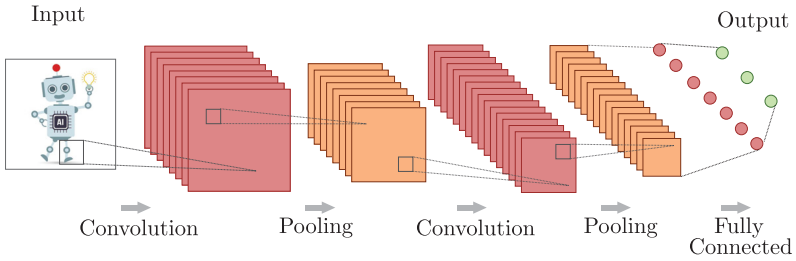


Figure 1.5. Model of a Convolutional Neural Network (CNN). The model takes as input an image. Then, it is alternatively composed of convolutional and pooling layers. Finally, Fully Connected (FC) layers are added for the classification task.

1.3.1 Convolutional layer

The convolutional layer is based on a mathematical operation called convolution. For example, for a two-dimensional image I and a two-dimensional

kernel K, the convolution is expressed as follows:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{1.17}$$

As the convolution is commutative, we can equivalently write:

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \tag{1.18}$$

In Deep Learning (DL), K, also called filter, is a weight matrix obtained by training the architecture. In practice, the operation implemented in DNNs is the correlation. It is the same operation except that the filter is not flipped. Given that the filter parameters are optimized during training, the flipping part is useless. We can consider that the optimized filter in a correlation implementation is the flipped version of the one that would have been computed if the convolution was implemented.

CNNs are composed of several convolutional layers. Each layer is composed of different filters. The output of a filter is called a feature map. Filters of the first layer have access to a small area of the input image, called the receptive field. However, the deeper the layer is in the network, the larger the receptive field grows as shown in Figure 1.6.

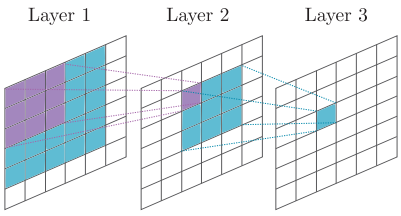


Figure 1.6. Visualization of the receptive field of a CNN. The neuron of Layer 3 has a receptive field of 3×3 in Layer 2 (in blue). As each neuron of Layer 2 has a receptive field of 3×3 in Layer 1 (in purple), the neuron of Layer 3 has a larger receptive field in Layer 1.

When CNNs are trained to solve a task, we observed that the weight matrices/filters are sensitive to some particular patterns [36]. Filters of first layers are sensitive to edge, texture or color. Deeper layers are sensitive to more complex patterns such as dog, bird, car, face, etc. (Figure 1.7). More complex filters would be located deeper in the network and would gradually be able to detect more sophisticated patterns.

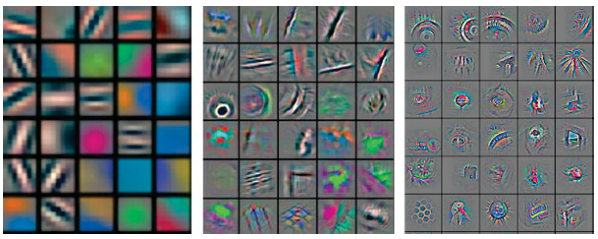


Figure 1.7. Visualization of the sensitive pattern for different layer levels in a CNN. On the left, these are simple patterns from the first layers of the network. In the middle, we see more complex patterns from the intermediate layers. On the right, these are the most complex patterns from the last layers of the network.

1.3.2 Pooling layer

The convolutional layer is usually followed by a pooling layer. The pooling layer is responsible for reducing the size of the convolutional layer output and decreases the computational power required to process the data. It is useful for extracting dominant features which are rotational and positional invariant.

They are two widely spread pooling methods: Max Pooling and Average Pooling. The Max Pooling returns the maximum value of the portion of the feature map covered by the kernel. On the other hand, the Average Pooling returns the average of all the values from the portion of the feature map covered by the kernel.

1.4 Recurrent Neural Networks

The architectures presented so far have only feedforward connections, meaning the information is transmitted through the network from the input to the output through a forward propagation. Nevertheless, it is possible to add to each neuron a self-looping connection and create a recursive behavior in the network. This kind of network is called Recurrent Neural Network (RNN) [24], a family of neural networks to process sequential data.

The main goal of RNNs is to take advantage of the sequential information. Similar to CNNs that can share parameters across space, RNNs can share parameters across time. The main difference with a Feedforward Neural Network (FNN) is the additional term due to the recurrent connection, the output depends on the previous state. Figure 1.8 shows an example along with the unrolled representation of the network throughout time.

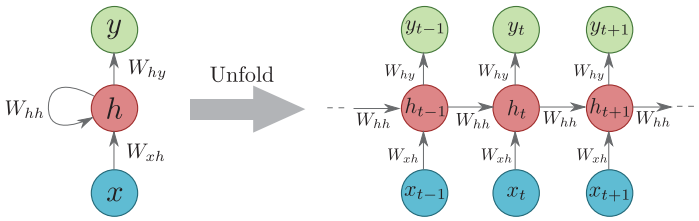


Figure 1.8. Model of a RNN along with the unrolled representation throughout time t. x is the input time sequence, h the hidden vector, y the output sequence and W the weights of the network.

Formally, given an input sequence $X = (x_1, \cdots, x_T)$, the RNN will compute the hidden vector $H = (h_1, \cdots, h_T)$ and the output vector sequence $Y = (y_1, \cdots, y_T)$ as follows:

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{1.19}$$

$$y_t = W_{hy}h_t + b_y \tag{1.20}$$

where f is an activation function, W_{xh} are the weights between the input and the hidden vector, W_{hh} are the weights between the two iterations of the same hidden layer, W_{hy} are the weights between the output and the hidden vector and b are the bias. The equation is applied from $t = 1$ to T.

Similar to FNNs, to train the network and update the weights W, the error is backpropagated from the upper layer to the lower layer but also from a hidden layer at the time step t to the time step $t-1$.

1.4.1 Unstable gradient problem with RNNs

RNNs face an issue with the gradient descent used to update the parameters [37]. Indeed, the gradient, propagated over many stages, tends to either vanish or explode. When computing the gradient in RNNs, the partial derivative $\frac{\partial h_i}{\partial h_{i-1}}$ is multiplied by itself several times due to the recurrence. Therefore, if $|\frac{\partial h_i}{\partial h_{i-1}}| < 1$, the gradient vanishes but if $|\frac{\partial h_i}{\partial h_{i-1}}| > 1$, the gradient explodes.

Specifically, whenever the model is able to represent long-term dependencies, the gradient of a long-term dependency has an exponentially smaller magnitude than the gradient of a short-term dependency. Therefore, it is not impossible to learn long-term dependencies, but it might take a very long time. Indeed, the long-term dependencies will tend to be hidden by the smallest fluctuations arising from short-term dependencies.

1.4.2 Long Short-Term Memory

A solution to the long-term dependency issue was found with Long Short-Term Memory (LSTM) cells [38]. LSTM networks are a special kind of RNNs capable of learning long-term dependencies. The classic recurrent cell is replaced by a more complex *memory cell*. It consists of additive gates through which the information flows. Equation 1.19 becomes:

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + b_i) \tag{1.21}$$
$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + b_f) \tag{1.22}$$
$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + b_o) \tag{1.23}$$
$$g_t = \varphi(W_{xg} * x_t + W_{hg} * h_{t-1} + b_g) \tag{1.24}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot g_t \tag{1.25}$$
$$h_t = o_t \odot \varphi(C_t) \tag{1.26}$$

where i_t, f_t, g_t and o_t are the input, forget, cell and output gates, respectively. x is the input sequence, h the hidden vector, C the memory cell, σ the sigmoid function, φ the Hyperbolic Tangent and $\odot$ the Hadamard product. W and b are the learnable weights and biases.

Figure 1.9 presents the diagram of a LSTM cell. The different gates aims to regulate the interactions between the LSTM cell and its environment. The forget gate (Equation 1.22) regulates the information of the memory cell, what information to forget or not. The input gate (Equation 1.21) decides what new information, from the cell gate (Equation 1.24), will be stored in the memory cell. The cell memory C_t is then updated by forgetting irrelevant information from the old cell memory C_{t-1} and adding new information from the cell gate g_t (Equation 1.25). Finally, the output gate (Equation 1.23) controls what information will come out (Equation 1.26).

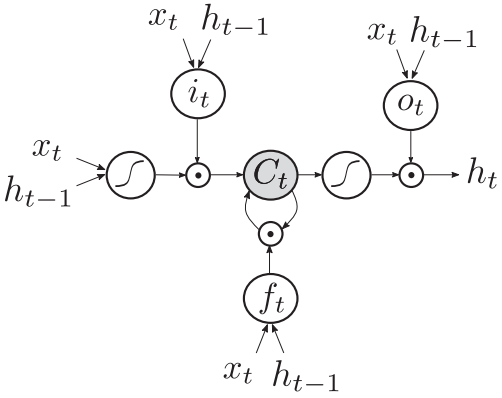


Figure 1.9. Model of a LSTM cell.

1.4.3 Gated Recurrent Unit

Many variants of LSTM can be designed by slightly changing the internal architecture. One of the most known is the Gated Recurrent Unit (GRU) [39]. Compared to the LSTM, a single gating unit, called update gate, combines the forget gate and the input gate. The GRU also merges the cell state and the hidden state. Equation 1.19 becomes:

$$z_t = \sigma(W_{xz} * x_t + W_{hz} * h_{t-1} + b_z) \tag{1.27}$$
$$r_t = \sigma(W_{xr} * x_t + W_{hr} * h_{t-1} + b_r) \tag{1.28}$$
$$\hat{h}_t = \varphi(W_{xh} * x_t + W_{hh}(r_t \odot h_{t-1}) + b_h) \tag{1.29}$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \tag{1.30}$$

where z_t and r_t are the update and reset gates, respectively. x is the input sequence, h the hidden vector, σ the Sigmoid function, φ the tanh and $\odot$ the Hadamard product. W and b are the learnable weights and biases.

The update gate helps the model to determine how much of the past information (from previous time steps) needs to be passed along to the future. The reset gate decides how much of the past information to forget.

1.4.4 Bidirectional Recurrent Neural Networks

The basic RNN has only access to past information to generate an output. In 1997, the bidirectional RNN is proposed in [40]. Instead of considering only the past, the past and the future are exploited simultaneously. The network has access to both upstream and downstream information of a sequence at every time step. In practice, the bidirectional RNN is composed of two independent RNNs. The input sequence is fed in normal temporal order for one RNN, and in reverse temporal order for the other. The outputs of the two networks are usually concatenated at each time step (Figure 1.10).

1.5 Attention Networks

Human beings are able to "pay attention" to some points, some parts of the surrounding environment. Attention is one step of perception: it analyses the outer real world and turns it into an inner conscious representation. When we look at a scene, an image, or a video to understand it, we focus on certain parts such as objects, peoples, actions, or even textures, colors, etc.

In computer vision, topographic maps, called saliency maps, are created to show and record where people look in an image. These maps are created with

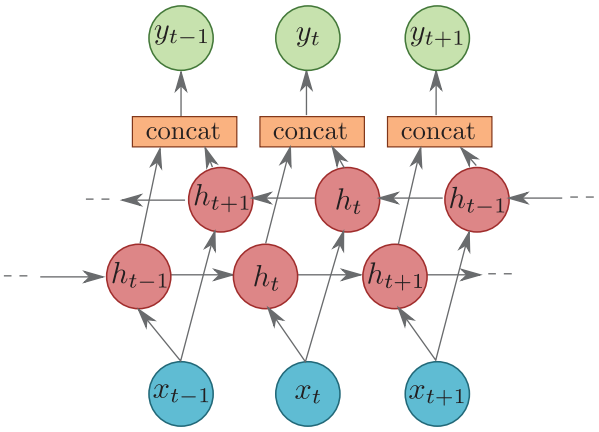


Figure 1.10. Model of a Bidirectional RNN. The input sequence x is fed into one "forward" RNN (from left to right) and one "backward" RNN (from right to left). The outputs are then concatenated to form the final output y.

eye-tracking or mouse-tracking. These data are usually used to train models that replicate the attention of the human being [12, 41].

Mimicking human attention may not be the main task, but a mechanism to solve various tasks. The model is not explicitly trained to focus on certain points. It learns by itself to pay attention to the relevant points while solving a specific task such as image classification [42, 43], neural machine translation [44, 45], image captioning [46, 47], etc. For example, the network pays attention to certain pixels of a image, called spatial attention, in the context of image classification [48]. The attention mechanism was proposed in [44] for Neural Machine Translation. The network focuses on certain words of a sentence (temporal attention).

1.5.1 Attention mechanism

The core idea of attention mechanisms is to focus on the most relevant parts of the input sequence or the input image for a given task. At each temporal step, a score is computed to determine the importance of each part of the input to solve the task. The score can be computed with different strategies.

Given a query q (encoded representation of the output) and a set of key-value pairs (k, v) (encoded representation of the input), the attention output y is a weighted sum of the values v. The weights α_{ij}, normalized with the softmax function, are dependent on the query q and the corresponding keys k:

$$y_j = \sum_i \alpha_{ij} v_i \tag{1.31}$$

$$\alpha_{ij} = \text{softmax}(e_{ij}) = \frac{e_{ij}}{\sum_k e_{kj}} \tag{1.32}$$

- Dot-Product Attention [49]:

$$e_{ij} = q_j^T k_i \tag{1.33}$$

- Scaled Dot-Product Attention [45]:

$$e_{ij} = \frac{q_j^T k_i}{\sqrt{d_k}} \tag{1.34}$$

- Multiplicative attention [49]:

$$e_{ij} = q_j^T W k_i \tag{1.35}$$

- Additive Attention [44]:

$$e_{ij} = W_3^T \tanh(W_1 k_i + W_2 q_j) \tag{1.36}$$

where W are weight matrices and vectors that can be learned during training. The query determines which values to focus on; we can say that the query 'attends' to the values.

Figures 1.11 and 1.12 show attention scores in the context of temporal attention for the translation task and spatial attention for the image captioning task, respectively.

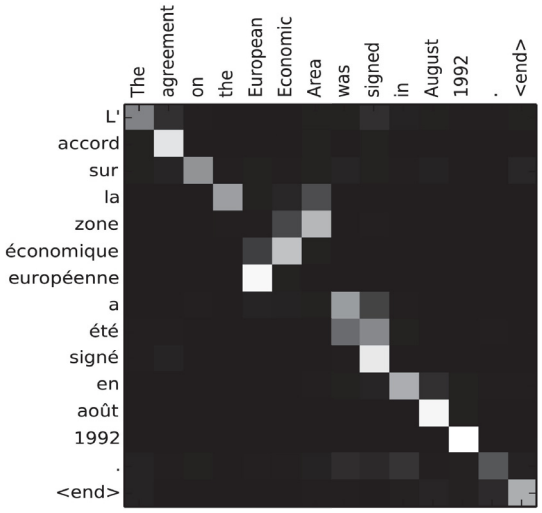


Figure 1.11. Attention score for a sentence in English and its translation in French. For each word in the translated sentence (lines), the white squares show the useful words in the input sentence (column). [44]

Figure 1.12. Visualization of attention maps. For each word of the generated caption, the relevant pixels determined by the attention mechanism to estimate the caption are highlighted. [46]

A particular case of the attention mechanism is the self-attention [50]. In this case, also known as intra-attention, the value, query and keys are encoded

representation of the input. The self-attention mechanism allows the input to interact with itself.

1.6 Normalization

Training networks with many layers can be difficult. Different techniques have been developed to facilitate the training of these networks such as normalization inside the model. We present two normalization techniques from the literature: batch normalization and layer normalization. Normalization is a technique for training DNNs that standardizes the inputs to a layer. This has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks.

1.6.1 Batch normalization

Batch normalization normalizes layer output in a network across the batch [51]. For each feature, batch normalization computes the mean and variance of that feature in the batch.

As a reminder, a batch is a set of examples used for an iteration of the learning algorithm. Given a batch composed of m examples, $B = \{x_1, x_2, \ldots, x_m\}$, $x_i \in \mathbb{R}^K$. The mean and the variance of the batch is computed by:

$$\mu_k = \frac{1}{m} \sum_{i=1}^{m} x_{ik} \tag{1.37}$$

$$\sigma_k^2 = \frac{1}{m} \sum_{i=1}^{m} (x_{ik} - \mu_k)^2 \tag{1.38}$$

Each sample is then normalized as follows:

$$\hat{x}_{ik} = \frac{x_{ik} - \mu_k}{\sqrt{\sigma_k^2}} \tag{1.39}$$

A mean of zero and a variance of one for each output, before the activation function, is not desired. Each sample is, therefore, scaled and shifted by

learnable parameters γ and β:

$$y_i = \gamma \hat{x}_i + \beta \tag{1.40}$$

During the testing stage, when the batch size is one, the mean and variance can not be computed. To overcome this problem, a "running mean" μ'_k and "running variance" σ'^2_k are updated in real-time during training and used during testing.

$$\mu'_k = \text{momentum} \times \mu'_k + (1 - \text{momentum} \times \mu_k) \tag{1.41}$$
$$\sigma'^2_k = \text{momentum} \times \sigma^2_k + (1 - \text{momentum} \times \sigma^2_k) \tag{1.42}$$

Batch normalization accelerates neural network training. However, it depends on the size of the batch, if the batch length is one, the variance is zero and batch normalization can not be applied. More generally, if the batch is too small, batch normalization makes the estimates very noisy and can negatively impact the training.

1.6.2 Layer normalization

On one hand, batch normalization normalizes the input features across the batch length. On the other hand, layer normalization normalizes the inputs across the features.

Given a batch of size m, $B = \{x_1, x_2, , \ldots x_m\}$, $x_i \in \mathbb{R}^K$, the layer normalization is expressed as follows:

$$\mu_i = \frac{1}{K} \sum_{k=1}^{K} x_{ik} \tag{1.43}$$

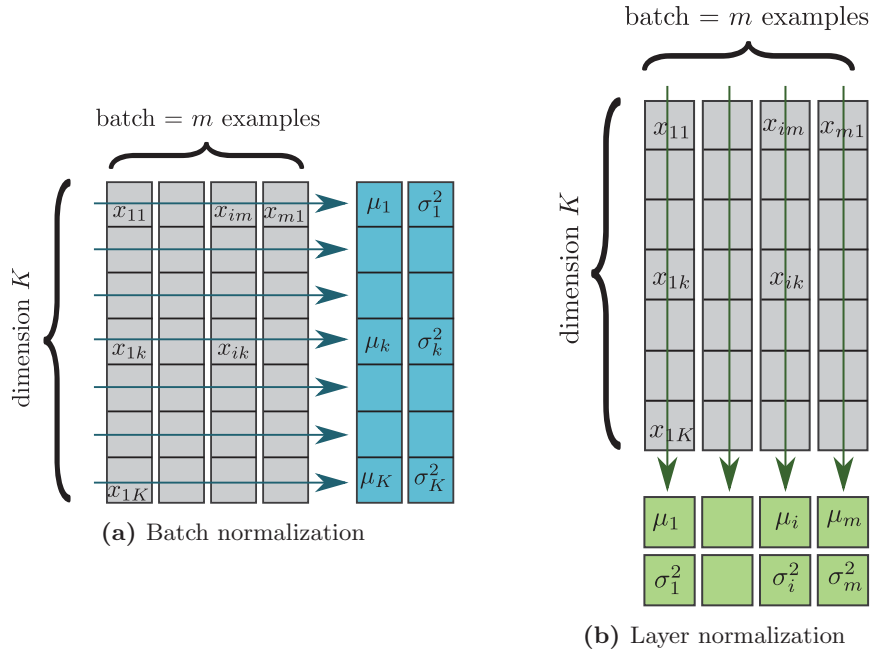
$$\sigma_i^2 = \frac{1}{K} \sum_{k=1}^{K} (x_{ik} - \mu_i)^2 \tag{1.44}$$

$$\hat{x}_{ik} = \frac{x_{ik} - \mu_i}{\sqrt{\sigma_i^2}} \tag{1.45}$$

As in batch normalization, each sample is scaled and shifted by learnable parameters γ and β:

$$y_i = \gamma \hat{x}_i + \beta \tag{1.46}$$

In batch normalization, the statistics are computed across the batch and are the same for each example in the batch. In contrast, in layer normalization, the statistics are computed across each feature and are independent of other examples (Figure 1.13).



(a) Batch normalization

(b) Layer normalization

Figure 1.13. Comparison between batch normalization (a) and layer normalization (b). With batch normalization, the mean μ and the variance σ^2 are computed across the examples inside the batch. With layer normalization, the statistics are computed across each feature.

1.7 In brief

Summary of Chapter 1

- In this chapter, we briefly introduced Deep Neural Network (DNN) principles. We started with the most simple model, the perceptron, and we followed with more complex architectures. We also presented the concepts of loss function and model learning.
- More specifically, we presented two main groups of DNNs: Feedforward Neural Networks (FNNs) and Recurrent Neural Networks (RNNs). In the FNN family, we introduced Multilayer Perceptrons (MLPs) and Convolutional Neural Networks (CNNs). CNNs are specialized for processing grid-like data such as image and spectrogram. We also explained RNN architectures such as Long Short-Term Memorys (LSTMs) and Gated Recurrent Units (GRUs), specialized to learn long-term dependencies.

Perspective for Chapter 1

- This chapter only presents a few DNN architectures. There are many architectures and a book would not be enough to describe every architecture proposed in the state of the art. DNNs evolve very quickly and offer increasingly complex solutions.
- This thesis focuses on the use of audio-visual data. The architectures presented in this chapter have only one input. In the next chapter, we will focus on models designed to handle jointly several inputs.

Chapter 2

Deep Neural Network with audio-visual data

Contents

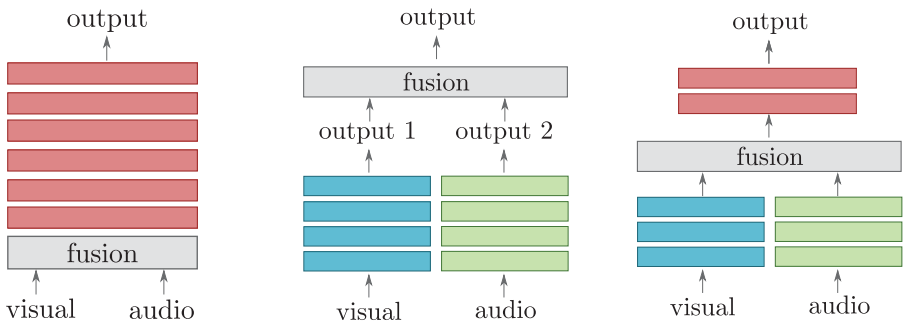
Living beings have different senses such as sight, hearing, touch, etc. A unique object or action can stimulate different modalities and therefore enrich the interpretation of the scene [1]. Among these numerous sensory streams, vision and audio are two modalities that simultaneously convey relevant information. In the case of audio-visual events, there are many examples such as the lip movement linked to the speech, the movement of fingers and the sound produced by the piano or even the movement of a car and the sound of an engine.

The audio-visual information has been used in numerous tasks such as sentiment analysis [52], emotion recognition [53], speech recognition [54], audio-visual separation [55, 56], audio-visual localization [57, 58], audio-visual correspondence learning [59], audio-visual generation [60], etc. However, few works in event classification exploit the information in both visual and audio modalities.

In this chapter, we present multimodal fusion techniques that are of general applicability. More precisely, we discuss different levels of fusion (at which level/depth of the DNN, the fusion is done). We also present fusion techniques. Finally, we introduce strategies to exploit simultaneously the visual and audio information proposed in different contexts.

2.1 Fusion levels

DNNs offer the flexibility of implementing multimodal fusion at different levels of the architecture. In the literature, the fusion techniques are classified into three strategies [61]: early, late and middle fusions as illustrated in Figure 2.1.



(a) Early fusion **(b)** Late fusion **(c)** Middle fusion

Figure 2.1. Illustration of the three fusion levels. Visual and audio inputs are fused before being processed by the network (a) or at the output of the network (b) or inside the network (c). Red blocks are multimodal layers while blue and green blocks are visual and audio layers, respectively.

2.1.1 Early fusion

The early fusion, also called feature fusion, is the fusion of the different modalities before entering the DNN (Figure 2.1a). The fused data are raw or preprocessed data from the sensors. This fusion is quite challenging because the data to be fused can be very different. The multimodal network is able to find correlations between the different modalities, but this technique may lead to very large input vectors with redundancies.

2.1.2 Late fusion

The late fusion, also called decision fusion, refers to the aggregation of decisions from multiple classifiers, each classifier is trained independently on separate modalities (Figure 2.1b). Various rules exist to combine the decisions of the classifiers: max-fusion, averaged-fusion, Bayes rule-based, or even learning using a metaclassifier. As one network is trained for each modality, this method lacks the correlation between the modalities, an important source of information. Besides, training individual models can be resource consuming.

2.1.3 Middle fusion

The middle fusion is a compromise between the early and late fusions. DNNs transform raw inputs to high-level representations by mapping the input through a pipeline of layers. It is, therefore, possible to fuse different representations into a single hidden layer and then learn a joint representation (Figure 2.1c). The different representations are fused using a fusion layer. The majority of DNNs adopts the middle fusion approach. We discuss different fusion techniques in the next section.

2.2 Middle Fusion techniques

Middle fusion can be implemented in different ways with DNN approaches. In this section, we describe several state-of-the-art fusion techniques [62].

2.2.1 Simple fusion techniques

The simplest way to fuse audio-visual features is the concatenation. Given the visual feature vector $v \in \mathbb{R}^{d_v}$ and the audio feature vector $a \in \mathbb{R}^{d_a}$, the audio-visual feature vector z is obtained by:

$$z = \text{concat}([v, a]) \tag{2.1}$$

Another simple fusion is to sum the two modality vectors to obtain an audio-visual representation, the sum is an element-wise addition. Vector sizes must be the same ($d_v = d_a$):

$$z = v + a \tag{2.2}$$

The additive approach can be more complex and implemented as a hidden layer in the DNN:

$$z = f(w_v^T v + w_a^T a) \tag{2.3}$$

where w are the weights connecting the modality layer to the shared layer and $f(.)$ is an activation function. In this case, there is no constraint on the vector size.

The last simple fusion is the multiplicative method by fusing visual and audio modalities with an outer product [63]. Each element of a modality is multiplied by each element of the other modality:

$$z = [v \otimes a] \tag{2.4}$$

where $\otimes$ indicates the outer product operator and [] denotes linearizing the matrix in a vector.

2.2.2 Multimodal Compact Bilinear Pooling

The multiplicative fusion leads to representations with very large dimensions. Gao *et al.* propose a more compact strategy called Multimodal Compact Bilinear pooling (MCB) [64–66]. The proposed method seeks to reduce the dimension of the outer product by Count Sketch projection (Equation 2.5).

Particularly, the count sketch projection of the outer product can be decomposed into a convolution of separated count sketch projections (Equation 2.6). Therefore, the computation of an outer product can be avoided. Finally, the authors use the FFT to compute a product instead of the convolution and accelerate the computation (Equation 2.7).

$$z = \Psi(v \otimes a) \tag{2.5}$$
$$= \Psi(v) * \Psi(a) \tag{2.6}$$
$$= \mathrm{FFT}^{-1}(\mathrm{FFT}(\Psi(v))) \cdot (\mathrm{FFT}(\Psi(a))) \tag{2.7}$$

where Ψ is the Count Sketch projection, $\otimes$ is the outer product and $*$ is the convolution.

2.2.3 Multimodal Factorized Bilinear Pooling

MCB usually needs high-dimensional features to guarantee robust performance, which may seriously limit its applicability due to limitations in GPU memory. To overcome this issue, Multimodal Factorized Bilinear Pooling (MFB) was proposed [67, 68].

Multimodal bilinear model can be expressed as:

$$z_i = v^T W_i a \tag{2.8}$$

where $W_i \in \mathbb{R}^{d_v \times d_a}$ is the projection matrix. $W = [W_1, ..., W_{d_z}] \in \mathbb{R}^{d_v \times d_a \times d_z}$ is used to get a d_z-dimensional output. However, this leads to a large number of parameters and high computational cost. Therefore, the authors propose to factorize W_i into two low-rank matrices, G_i and H_i:

$$z_i = v^T G_i H_i^T a = \mathbf{1}^T \left(G_i^T v \odot H_i^T a\right) \tag{2.9}$$

where $G_i \in \mathbb{R}^{d_v \times k}$ and $H_i \in \mathbb{R}^{d_a \times k}$ are the factorized matrices, k is the latent dimensionality, $\odot$ is the element-wise multiplication of two vectors, $\mathbf{1} \in \mathbb{R}^k$ is an all-one vector.

To obtain the output feature $z \in \mathbb{R}^{d_z}$, the weights to be learned are two three-order tensors $G = [G_1, \ldots, G_{d_z}] \in \mathbb{R}^{d_v \times k \times d_z}$ and $H = [H_1, \ldots, H_{d_z}] \in \mathbb{R}^{d_a \times k \times d_z}$. Without loss of generality, they can be reformulated as 2D matrices $G' \in \mathbb{R}^{d_v \times k d_z}$ and $H' \in \mathbb{R}^{d_a \times k d_z}$. Equation 2.9 can be rewritten as follows:

$$z = \text{SumPooling}(G'^T v \odot H'^T a, k) \tag{2.10}$$

where the function $\text{SumPooling}(x, k)$ means using a one-dimensional non-overlapped window with the size k to perform sum pooling over x and $\odot$ refers to element-wise multiplication.

2.2.4 Dual Multimodal Residual Fusion

The Dual Multimodal Residual (DMR) Fusion is inspired by the residual connection [69]. The audio and visual features are simultaneously updated by preserving the useful information in the original modality and by adding complementary information from the other modality [70]:

$$a' = \varphi(a + g(a, v)) \tag{2.11}$$
$$v' = \varphi(v + g(a, v)) \tag{2.12}$$

where $g(.)$ is an additive fusion function composed of dense layers (Equation 2.3) and φ is the Hyperbolic Tangent (tanh).

2.3 Audio-Visual Learning

This manuscript focuses on two modalities: visual and audio data. Research in the simultaneous contribution of visual and audio information is being conducted since the last decades [71].

For example, the lip movement is used in addition to speech for audio-visual automatic speech recognition. The first automatic speech reading system was proposed by Petajan [72]. Then, Yuhas *et al.* proposed to use both modalities together [73]. Since then, many studies have investigated different strategies to

jointly exploit audio-visual information [54, 74–77]. Several studies have been done, for example, the comparison of early and late fusion strategies [78].

The fusion of visual and audio information is also beneficial to emotion recognition. Basically, emotion can express through several social behaviors, including facial expression, speech, text, gesture, etc. Among these modalities, facial expression and speech are important and natural channels to transmit human affective states. Several works have investigated the combination of visual and audio information in this context [53, 79].

Visual and audio information is also jointly used in the context of audio-visual correspondence learning. For example, given facial images and the corresponding audio sequences, voice-facial matching aims to identify the face that the audio belongs to or vice versa [57, 80, 81]. Arandjelovic *et al.* introduced an audio-visual correspondence learning task [82], meaning finding the image that corresponds to the sound. Several works continued to investigate and focus on finding the most similar visual area to the current audio clip [83–86]. Korbar *et al.* introduced a similar task called audio-visual temporal synchronization [87] which determine whether a given audio sample and video clip are "synchronized" or "unsynchronized". [88].

Finally, audio-visual information was also exploited in the context of cross-modal learning. The learning of models is carried out following a student-teacher perspective. For example, a state-of-the-art network for vision teaches the sound network to recognize scenes and objects [89] or the opposite [90].

2.4 In brief

Summary of Chapter 2

- In this chapter, we first introduced different fusion levels present in the literature: early, late and middle fusions.
- In the context of Deep Neural Networks (DNNs), the most used fusion level is the middle fusion as the fusion can be included directly into the network as a hidden layer. We presented different fusion techniques from the literature: concatenation, additive fusion, multiplicative fusion, Multimodal Compact Bilinear pooling (MCB), Multimodal Factorized Bilinear Pooling (MFB), Dual Multimodal Residual (DMR).
- Finally, we focused on the use of audio-visual data in the DNN literature and noticed the positive impact of using jointly audio and visual information in different areas.

Perspective for Chapter 2

- Despite the advances made on multimodal models, they are still limited to restricted areas. Studies on the fusion of audio-visual information in the context of event classification and localization are rare. Audiovisual models for event classification are discussed in detail in Chapter 7.

Part II

Databases

Chapter 3

Related Work

Contents

Annotated and large datasets are required for supervised training of Deep Neural Networks (DNNs). We were not able to find sets that suit our needs. In this chapter, we review datasets that are close to the event classification and localization problem. We first present datasets based only on sound information for the particular tasks of sound event classification and sound event localization. Then, we focus on datasets based on visual information for video classification and event localization. Finally, we present multimodal datasets for audio-visual event classification.

3.1 Sound datasets

The presented sound datasets are divided into 3 types depending on the task at hand: Sound Event Detection (SED), Sound Source Localization (SSL), and Sound Event Localization and Detection (SELD). The SED goal includes recognizing each sound event class present in acoustic scenes and locate each event in time. On the other hand, SSL goal is to locate in space each event without classifying them. SELD is the combination of the two tasks.

3.1.1 Sound event detection and classification

As supervised learning methods require large sets of annotated data, datasets for sound event classification were created such as AudioSet [91], ESC-50 [92] or CURE [93]. AudioSet was created to accelerate research in the area of acoustic event classification, just as ImageNet has driven research in image understanding. The dataset is a large-scale weakly labeled collection of 10 second long audio recordings extracted from about 2 million YouTube videos. ESC-50 and CURE are smaller dataset, queried from Freesound online repository.

These datasets have only few metadata. More complete datasets were created to incorporate temporal information for sound event detection such as Urban Sound [94], TUT Sound events [95] and VOICe [96]. They are labeled with the beginning and end of the events as well as the event class. However, these datasets do not comprise any information about localization in space.

In the context of the Computers in the Human Interaction Loop (CHIL) project, CHIL-UPC and CHIL-ITC datasets were recorded [97]. Both datasets include sets of isolated acoustic events that occur in a meeting room environment. The acoustic scenes were recorded with multiple microphones. The UPC dataset includes 13 semantic classes with around 60 examples per sound classes. The participants took a different place in the room out of 7 fixed different positions. The ITC dataset includes 16 semantic classes of events with around 50 sounds per almost each of the sound classes. Since the purpose of these datasets was to collect real data for the detection problem, they have few different positions in the room.

3.1.2 Sound source localization

Some datasets were created for sound event localization such as MUltimicrophone Source LOcalization Database (MUSLOD) [98] or AV16.3 [99]. In MUSLOD, the events are situated at a fixed distance from the microphones with a variable angle of incidence. In AV16.3, there is a greater variability of positions. Furthermore, the event localization is made in the 3D space and in 2D images of the scene. However, the events are always speech in both datasets.

3.1.3 Sound event detection and localization

The Single- and Multichannel Audio Recordings Database (SMARD) [100] includes the class and the position in space of the event. 48 configurations are possible by changing the position of the sources, the position of the sensors, the type of loudspeakers and the type of sensors. However, the events are sequences played on loudspeakers and no event overlap is present in the dataset.

Most of the time, the researchers evaluate the performance of SELD models on simulated data as in [101]. The reverberation of the room can be simulated with different techniques. The most realistic dataset were created for the DCASE2019 challenge [102]. Real-life Impulse Responses (IRs) are collected from 5 indoor environments. In each room, the real-life IR is recorded at 504 unique combinations of azimuth-elevation-distance. Then, the collected IRs are convolved with isolated sound events from the DCASE2016 task. However, each event class can occur anywhere in the room, even in unrealistic locations.

Table 3.1 summarizes the content of Section 3.1.

	Duration	Classes (Type)	Time	microphones	Location	overlap	Real data	Available online
AudioSet [91]	5,800h	527 (general)		1		✓	✓	✓
ESC-50 [92]	2.78h	50 (general)		1			✓	✓
CURE [93]	9h	13 (general)		1			✓	
Urban Sound [94]	8.75h	10 (Outdoor)	✓	2			✓	✓
TUT Sound events [95]	1.3h	18 (Outdoor, Indoor)	✓	2		✓	✓	✓
VOICe [96]		3 (baby crying, glass breaking, gunshot)	✓	1		✓		✓
UPC [97]		13 (Office)	✓	84	7		✓	
ITC [97]		16 (Office)	✓	32	4		✓	
MUSLOD [98]	18h	1 (Speech)		4	11			
AV16.3 [99]	1.42h	1 (Speech)		16	16	✓	✓	✓
SMARD [100]		20 (Artificial, Speech, Musical)		1 to 24	8			✓
simulated data [102]	6.67h	11 (Office)	✓	4	504	✓		✓

Table 3.1. Comparison of sound datasets according to different criteria: total duration, number of classes, presence of temporal information, number of microphones, presence of location information, presence of overlap between events, data realism and online availability.

3.2 Visual datasets

Numerous datasets are composed of visual information for event classification and/or localization. Several examples are given, but this list does not represent all existing datasets.

3.2.1 Visual event detection and classification

Several datasets, composed of images, were created for object classification and/or detection (ImageNet [103], PASCAL VOC [104], COCO [105], etc.). Temporal information is relevant information for event classification and detection, but it is not present in these datasets.

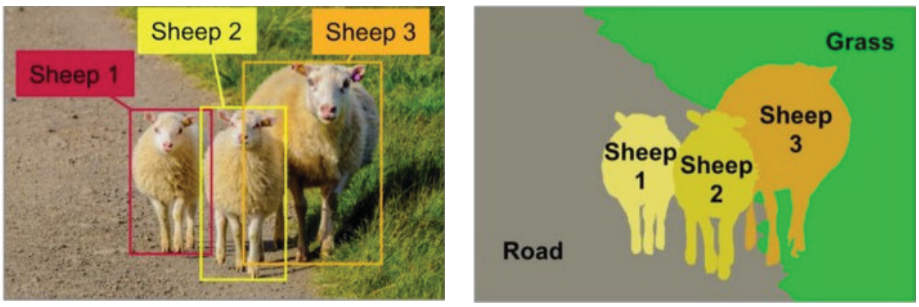
Most event datasets were created by taking videos from YouTube and by annotating manually and/or automatically the videos (YouTube-8M [106], Kinetics-700 [107], HMDB [108], UCF101 [109], Sports-1M [110], etc.). Other datasets were collected according to specific scripts or purposes such as KTH [111] and Something-Something [112].

In most cases, datasets are composed of several classes representing a wide variety of human activities, but some datasets are more related to a specific topic. For example, Sports-1M [110] focuses on the classification of multiple sports, epic-kitchens [113] on egocentric kitchen-related action, etc.

Finally, all these datasets are only annotated with class information for the entire video, but more complex datasets exist such as Charades [114], THUMOS [115], ActivityNet [116], etc. These datasets also include temporal information, the beginning and the end of each event.

3.2.2 Visual event localization

The researchers extended the object localization in images to localization in videos, by annotating each frame. Events can be located in each frame with bounding boxes as illustrated in Figure 6.2a (YouTube-BoundingBoxes [117], epic-kitchens [113], AVA [118]). On the other hand, a more complex strategy is to associate a class to each pixel as illustrated in Figure 6.2b (VOS [119] and DAVIS [120]).



(a) Bounding boxes **(b)** Segmentation at pixel level

Figure 3.1. Localization of objects/actions. On one hand, the objects are outlined in boxes and associated with a class (a). On the other hand, a class is associated with each pixel (b).

To our knowledge, unlike sound datasets, there is no localization of events or actions with spatial coordinates.

Table 3.2 summarizes the comparison of the different visual datasets.

	Duration	Classes (Type)	Multilabel	Time	Bounding Boxes	Segmentation
YouTube-8M [106]	350,000h	3862 (general)	✓			
Kinetics-700 [107]	1,800h	700 (human action)				
HMDB [108]	6849 clips	51 (human action)				
UCF101 [109]	27h	101 (human action)				
Sports-1M [110]	100,000h	487 (sports)	✓			
KTH [111]	600 videos	6 (human action)				
Something-Something [112]	245h	174 (human action)				
Charades [114]	9848 videos	203 (human action)	✓	✓		
THUMOS [115]	430h	101 (human action)		✓		
ActivityNet [116]	648h	200 (human action)	✓	✓		
epic-kitchens [113]	740h	456 (kitchen activity)	✓	✓	✓	
YouTube-BoundingBoxes [117]	240,000 videos	23 (general)			✓	
AVA [118]	107h	80 (human action)	✓	✓	✓	
VOS [119]	5.5h	94 (general)	✓			✓
DAVIS [120]	2.5 min	1 (foreground)				✓

Table 3.2. Comparison of visual datasets according to different criteria: total duration, number of classes, presence of several classes in one video, presence of temporal information, localization with bounding boxes and localization by segmentation.

3.3 Audio-visual datasets

In recent years, researchers have decided to exploit the audio information, included in videos, in addition to visual information. Most datasets based on videos include a soundtrack. However, as the dataset was annotated based only on the visual information, the soundtrack does not always include relevant information. Moreover, these datasets comprise classes that do not produce a particular sound signature such as shake hands, push up, etc. Some videos also include irrelevant background noise, for example, background music added to the video.

However, some audio-visual datasets have recently been created. Moments In Time (MIT) dataset [121] are composed of actions and events that may be visual and/or audible. Labels are verbs and present a wide variety of examples for the same label. MIT has a significant intra-class variation among the categories. For example, "playing" may includes many action categories such as "playing guitar", "playing a video game" or "playing in the garden". In other datasets, these actions are labeled with separate labels. MIT have recently been enlarged to the Multi-Moments in Time [122]. In the new dataset, a video can comprise several labels.

Tian *et al.* create the AVE dataset for audio-visual event detection [70]. It covers a wide range of audio-visual events from different domains, e.g., human activities, animal activities, music performances, and vehicle sounds. The dataset is built based on visual and audio information. The event is present if it is both visible and audible.

VGGSound [123] was collected from YouTube using image classification algorithms. Videos with irrelevant background sound are then filtered out. A VGGish model with only three sound classes (speech, music and others) is used to exclude videos including a narrator describing it or background music. VGGSound ensures audio-visual correspondence and is collected under unconstrained conditions. Categories cover a wide variety of events that can be grouped as people, animals, music, sports, nature, vehicle, home, tools, and others.

Smaller audio-visual datasets have been recorded for a specific application, for example, the classification and detection of human manipulation actions [124].

All these databases do not include location information. Indeed, events are not located in the different frames of the video, there is no bounding boxes or segmentation map provided. Moreover, as the events can take place both outside and inside, it is also not possible to provide x,y coordinates to locate the events in a room.

Table 3.3 summarizes the different audio-visual datasets.

	Duration	Classes	Multilabel	Time
Moments In Time [121]	833h	339 (Verb)		
Multi-Moments in Time [122]	833h	339 (Verb)	✓	
AVE [70]	11.5h	28 (general)		✓
VGGSound [123]	200,000 videos	300 (general)		
Human manipulation actions [124]	8 videos	6 (sub-actions)		✓

Table 3.3. Comparison of audio-visual datasets according to different criteria: total duration, number of classes, presence of several classes in one video and presence of temporal information.

3.4 In brief

Summary of Chapter 3

- In this chapter, we first presented different sound datasets in the context of event classification, detection and localization. On one hand, the datasets comprise different labels but do not include information about the event localization in space. On the other hand, different annotated locations are provided but the datasets are only composed of one event: speech. The only dataset that comprises several labels and different locations in space was created using the convolution of real sounds with real Impulse Responses (IRs).
- We then presented different datasets based on visual information. There are numerous visual datasets, they usually include different classes and many examples which do not facilitate the interpretation of results. Moreover, the localization of the events are made directly in the frames of videos. None of these datasets locates events in space.
- Finally, we focused on audio-visual datasets. These datasets are fewer. They have a better audio-visual correspondence compared to visual datasets. Indeed, visual datasets include sometimes soundtrack, but it does not necessarily correspond to the image.

Perspective for Chapter 3

- Only few databases were created taking into account visual and audio information simultaneously. Furthermore, these databases are very general. They include a wide variety of categories. They are poorly suited for the evaluation of scene analysis models, for example, models performing the classification and localization in space.
- Moreover, as there are no constraints on the events, they occur at various indoor and outdoor places. Therefore, it is not possible to localize them in space.

Chapter 4

Data collection

Contents

This chapter is based on the following publication:

- Mathilde Brousmiche, Jean Rouat, Stéphane Dupont. "SECL-UMons Database for Sound Event Classification and Localization". In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

In this chapter, we present the new audio-visual event dataset recorded in the context of office environments. It fills a gap in the currently available datasets. It includes audio-visual recordings as well as metadata such as the event class and location in a room. The dataset was designed to evaluate scene understanding models, more precisely models that classify and localize events based on visual and audio information.

The dataset is available in the form of two folders: **SECL-UMONS**[1] includes recordings of several events with a microphone array and **AVECL-UMONS**[2] includes recordings of the same events with four webcams. Both folders are available on Zenodo.

4.1 Recording conditions

Audio-visual events were recorded with 4 Logitech C920 webcams[3] (AVECL-UMONS) and a UMA-8-SP microphone array[4] (SECL-UMONS).

OBS software was used to capture the webcam streams with 20 frames per second and frame dimensions of 1920×1080. The webcams include stereo microphones (sampling rate of 44.1 kHz). The four webcams were placed in the four corners of the room. Each event is in the field of view of at least one camera.

The circular array is made of seven omnidirectional microphones and was placed at the center of the room. The microphone streams were captured by Audacity software with a sampling rate of 44.1 kHz and 32-bit encoding.

[1] https://zenodo.org/record/3965492#.X1DKOobgrCI
[2] https://zenodo.org/record/3932885#.X1DKPobgrCI
[3] https://www.logitech.com/en-us/product/hd-pro-webcam-c920
[4] https://www.minidsp.com/products/usb-audio-interface/uma-8-sp-detail

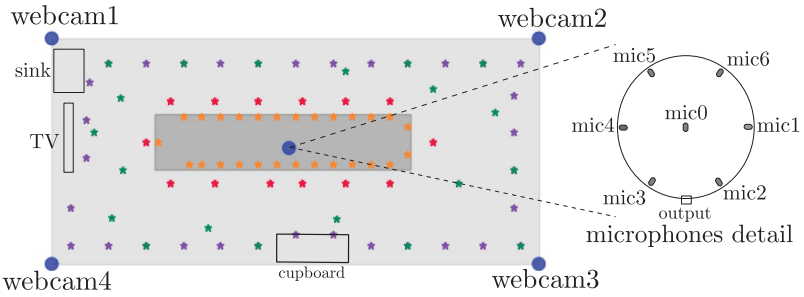
4.2 Dataset description

The dataset comprises 11 classes, each having several subclasses. The difference between subclasses is either the use of a different object belonging to the same class or a different participant performing the action (Table 4.1).

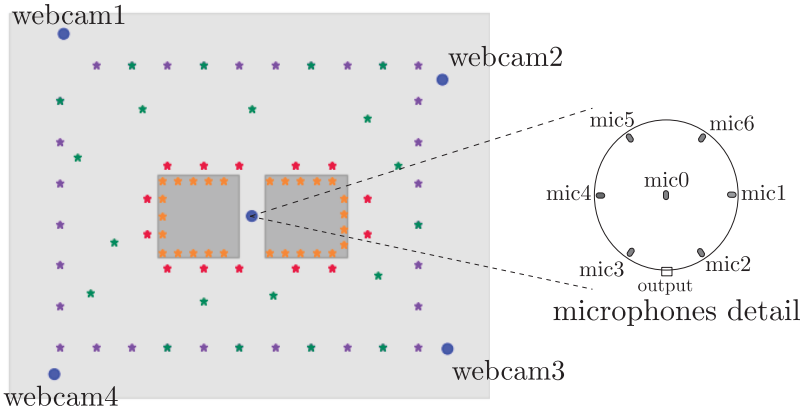
Events were recorded in two different rooms. Depending on the event classes, different positions were possible in the room (Figure 4.1). For each subclass, an event was recorded at each possible position.

Class	# of subclasses	# of possible positions in Room 1	# of possible positions in Room 2
chair_movement	4	14	14
cup_drop_off	4	27	27
hand_clap	4	34	34
keyboard	4	14	14
knock	2	32	33
phone_ring	6	27	27
radio	4	33	33
speaker	4	34	34
step	4	39	43
whistle	4	34	34
furniture_drawer	4	29	33

Table 4.1. List of class with respective number of subclasses, possible positions in room 1 and 2.



(a) Room 1



(b) Room 2

Figure 4.1. Diagram of Room 1 and Room 2. The blue dots are the positions of the webcams and the microphone array. The other dots are the possible positions in the room for the different event classes. Orange: Cup drop off, Keyboard, Phone ring; Red: Chair movement, Hand Clap, Speaker, Whistle; Green: Hand Clap, Speaker, Step, Whistle; Purple: Furniture's drawer, Knock, Step.

4.3 Recording Process

To avoid the demanding step of manual annotation, a recording process was created. Events were realized at specific positions and specific times following a predefined scenario. Sequences of interest were then extracted and auto-

matically annotated. For each class, several positions were marked in the two rooms before starting the recordings.

The dataset is divided into two parts according to the number of events in the sequence: unilabel sequences (one event per sequence) or multilabel sequences (two simultaneous events per sequence).

4.3.1 Unilabel sequences

Recordings of several minutes, named session, were saved. Afterwards, the sequences of interest (periods of time in which audiovisual events take place) were extracted. One session was realized for each subclass. During a session, the participant realized the event at each possible position marked beforehand. A script, shown on a screen in the room, was run for each session. When and where the events had to occur were ordered by this script. To avoid the presence of the noise of the participant movement in final sequences, when to move between two events was also ordered by the script. Afterwards, sequences of interest were automatically extracted from the session recordings with the time planned by the script. A total of 2662 sequences of 3 seconds composed of only one event were extracted from the different session recordings.

Figure 4.2. Example of unilabel data for each webcam in Room 1.

4.3.2 Multilabel sequences

Multilabel sequences are composed of two event classes realized approximately at the same time. The combinations of two event classes were chosen for all possible duos of events. Each event class (except *furniture_drawer*) was associated with all classes, even with itself. 55 possible associations were created. A session was realized for each duo of event classes. Both participants take different positions in the room. 25 positions were selected to cover a maximum of situations (all around the room, away from each other and close from each other). Again, a script was run for each session. When and where the events had to occur as well as when to move to avoid parasitic noise were ordered by this script. Afterward, sequences were extracted from the session recordings with the time planned with the script. A total of 2729 sequences of 4 seconds composed of two events were extracted.

Figure 4.3. Example of multilabel data for each webcam in Room 1.

4.4 Metadata

Different metadata are provided with the unilabel and multilabel sequences. For each unilabel sequence and for each label present in each multilabel sequence, the following information is provided:

- event class;
- event subclass number;

- x,y,z coordinates in the room;
- number of the room;
- event presence or not in the field of view of each webcam.

4.5 Task setup

Following the split of data into test and training sets, it is possible to evaluate different aspects of the models (for example, the model ability to generalize with a subclass never seen during training). Therefore, two different splits of the data into training and test sets are proposed for the unilabel sequences:

- Split1: a classical random split of data. 44 sequences are chosen randomly in each class (22 in each room) to constitute the test set. The rest of the sequences constitutes the training set. A total of 484 sequences is used for the test set and 2178 sequences for the training test.
- Split2: a split aiming to test the generalization ability of the model. One subclass of each class is used as test data. A total of 671 sequences constitutes the test set and 1991 sequences the training set.

For multilabel sequences, two splits of the data into training and test sets are also proposed. The second split tests the ability to classify a combination of classes never done during training:

- Split1: 10 sequences are chosen randomly for each possible duo between classes (5 in each room) to constitute the test set. The rest of the sequences constitutes the training set. The test set includes a total of 550 sequences and the training test includes 2179 sequences.
- Split2: 10 duos of event classes are chosen to constitute the test set. The rest of the sequences constitutes the training set. The test set and training set are composed of 500 and 2229 sequences, respectively.

4.6 In brief

Summary of Chapter 4

- In this chapter, we presented a novel dataset to evaluate models for audio-visual event classification and localization with a sufficient amount of data for neural network training.
- The new dataset includes:
  - recordings from a microphone array and four webcams;
  - 11 event classes from real-life office environments;
  - a total of 5.24 hours of recordings
  - several possible realistic positions in two different rooms
  - 2 types of sequences: unilabel and multilabel.

Perspective for Chapter 4

- The database does not currently allow detection because no accurate temporal information about events is available. Indeed, despite the recording protocol, the person does not execute the action at the precise moment ordered by the script and the duration of each action is not known. Therefore, an added bonus for the dataset would be to manually annotate the beginning and end of each event.

Part III

Sound event classification and localization

Chapter 5

Related work

Contents

Sound Event Localization and Detection (SELD) can be divided into two subtasks: Sound Event Detection (SED) and Sound Source Localization (SSL). The SELD goal includes recognizing each sound event class present in the acoustic scene and simultaneously locate in space each detected sound event (Figure 5.1). For many years, SED and SSL have been evaluated separately.

In this chapter, we present state-of-the-art architecture based DNNs for SED, SSL and finally SELD.

5.1 Sound Event Detection

In recent years, different models have been tested in the literature to address Sound Event Detection (SED) problem. The first ones were based on MLPs [125, 126]. Different input features such as mel-band energies, log mel-band energies and Mel-Frequency Cepstral Coefficient (MFCC) were proposed [125]. The comparison between multiple classifiers, one for each class, and one multi-class classifier was studied in [126].

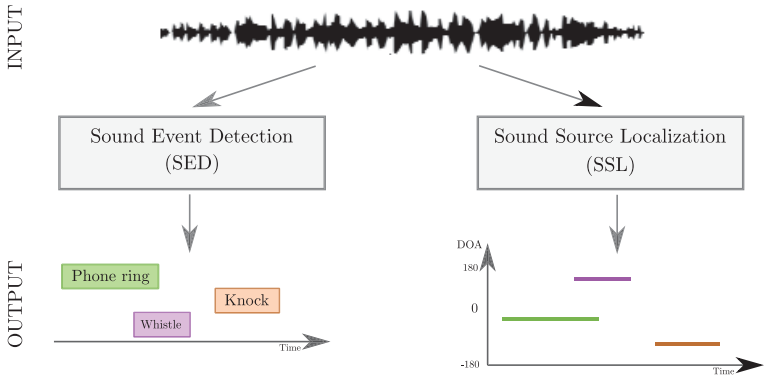


Figure 5.1. Sound Event Localization and Detection (SELD) task composed of the Sound Event Detection (SED) and Sound Source Localization (SSL) subtasks. Given an acoustic scene (input), the SED model estimates the beginning and end of each event as well as the class. The SSL estimates the localization in space of each event.

The CNN, known to extract the high-level features that are invariant to local spectral and temporal variations, was also used for SED [127,128]. Zinemanas *et al.* proposed an end-to-end CNN that does not require feature extraction such as spectrogram, mel-band energies or MFCC [129]. Their model includes a 1D CNN (which computes features similar to mel–frequency bands from the raw audio signal), followed by a 2D CNN for the classification task.

As RNNs are suitable for input data with sequential structure, they were used to adress the SED problem [130–134]. Valenti *et al.* made a comparison between MLP and RNN with different monaural and binaural audio features as input such as log mel-band energies and MFCC [135]. Bidirectional LSTM and GRU, powerful techniques to exploit context information from the past and the future, were also studied in [136–138].

More recently, researchers have proposed to take advantages of both approaches. They implemented a combination of CNN and RNN named Convolutional Recurrent Neural Network (CRNN) [139–142]. CRNN outperformed all previous SED methods. Adavanne *et al.* went further by using 3D CNN instead of 2D CNN to simultaneously learn the inter- and intra-channel features from multichannel audio input [143].

As some datasets were recorded with a microphone array, some works tried to exploit a maximum of the available information. For example, the input may be composed of several spectrograms, one for each microphone channel [132]. More complex features were also computed such as Time Difference of Arrival (TDOA) [131, 133] or Generalized Cross Correlation (GCC) [143]. They were usually used with spectrograms and/or mel-band energies.

On the other hand, researchers proposed to include capsules in CNNs or CRNNs for SED [144–147]. A capsule is a set of neurons that activate for various properties such as position, size and hue. The capsule was introduced by Hinton in [148]. The output of one capsule is proportional to the presence of one specific entity inside the acoustic scene. The second level of capsules includes a composition of the previous layer and represents an object composed of lower level capsules. The relationship between capsule levels is regulated by a routing mechanism. The connection between lower level capsules and the next level capsules is stronger when the presence of the children implies the presence of the parent in the scene. The connection decreases when there is no relationship between parent and child. So researchers proposed to exploit the capsule units to represent a set of distinctive properties for each individual sound event.

All these works estimate classes for each frame of the acoustic scene. However, some researchers tested to estimate classes at event level directly. Inspired by Faster-RCNN [149] for image detection, researchers proposed to generate event proposals (temporal intervals) for SED [150–154]. Several temporal regions, where an event may be present, are proposed by a region proposal network. These regions are then fed into a final network to estimate the event class and refine the center and length of the time interval.

Finally, researchers also proposed to add attention mechanisms to improve SED performances [155, 156]. The proposed models learn when to listen using temporal attention and where to listen on the frequency axis using frequential attention.

5.2 Sound Source Localization

There are many techniques to estimate the event location in space. In the last years, different methods based on neural networks have been studied. The main differences between the different neural network proposals reside in the input features, architectures and network output (target).

The input features of the network are frequency domain features [157–161], features derived from the GCC [162,163], acoustic intensity vector [164,165] or even raw sound [166, 167]. As the phase of the spectrogram includes relevant information for the localization of sound events, some researchers designed a novel model that can handle complex numbers [168, 169] as input. Other researchers took as input the amplitude and the phase of the spectrogram [157, 158].

The neural network architectures are similar to architectures created for SED such as MLPs [168, 170], CNNs [157, 161, 166, 171], RNNs [172] or CRNNs [158–160, 164, 165].

The network output can be expressed in several ways. On on hand, the localization is achieved by estimating the Direction of Arrival (DOA) through classification [157, 158, 161, 162, 168, 171]. On the other hand, the position coordinates are estimated trough regression [163, 166] or through classification with a predefined spatial grid [173]. Some researchers also proposed to apply multi-task learning. For example, one output estimates the azimuth and the second output estimates the elevation [174]. In [170, 175], the first task is to determine the number of sources and the second task is to estimate the DOA for each source.

5.3 Sound Event Localization and Detection

Recent research attempts to solve the two subtasks as a joint task. DCASE2019 proposed for the first time the SELD task in their challenge and a total of 58 systems were submitted [176]. In the baseline system [101], each input frame is mapped with a CRNN into two parallel outputs. The first one performs the sound detection, by classifying the active sound event class. The second

one estimates the localization of the detected sound event with a multi-class regression.

Most models proposed for the challenge were CRNN [177–179], only few models were composed of CNN without recurrent layers [180, 181]. As previously, the researchers tried different inputs such as phase and magnitude spectra, GCC, intensity vector, log mel-band energies, etc. Most of the time, several features were used conjointly. Two different strategies were implemented to address the two tasks simultaneously, either they used multi-task learning as the baseline model [179, 182, 183] or they trained two or more distinct models [178,184]. For example, the model, with the best result for the DCASE2019 challenge [177], was composed of four CRNNs. One task was associated to each CRNN: the source number estimation, the DOA estimation of the active source, the DOA estimation of the second source in the case of two simultaneous events were detected and the classification of the events. Some researchers also proposed to use parametric DOA estimations [179,184]. Finally, data augmentation was applied to improve the result of some models in [185–187].

Other techniques were studied without being submitted to the challenge. For example, Comminiello *et al.* proposed a quaternion convolutional neural network [188]. The key point was that the convolution process was performed in the quaternion domain (a number system that extends the complex numbers). The CRNN was slightly modified in [189]. On the other hand, the CNN was replaced with a U-Net (composed of convolutional and deconvolutional layers). Sound event detection mainly relies on time-frequency patterns while DOA estimation relies on magnitude or phase differences between microphones. Therefore, the Sound Event Detection and the DOA estimations were decoupled in the first step of the network in [190]. The information was then merged at the level of the recurrent layer. Finally, instead of recurrent layers in the CRNN, Guirguis *et al.* proposed to use Temporal Convolutional Network (TCN) to decrease the inference time [191].

5.4 In brief

Summary of Chapter 5

- In this chapter, we presented state-of-the-art models for Sound Event Detection (SED), Sound Source Localization (SSL) and the combination of both tasks Sound Event Localization and Detection (SELD).
- For SED, different models were explored such as MLP, CNN, RNN and the combination of CNN and RNN named Convolutional Recurrent Neural Network (CRNN). Inspired by the computer vision, networks composed of capsules or temporal region proposal were also studied.
- For SSL, researchers proposed similar architectures (MLP, CNN, RNN and CRNN) but also explored different input features (frequency domain features, GCC, acoustic intensity vector, etc.). They also formulated the problem in different ways: Direction of Arrival (DOA) estimation with classification or position coordinates estimation with regression.
- Finally, to combine both tasks into a single problem (SELD), either multi-task learning was used or several networks were trained.

Perspective for Chapter 5

- Most works have been evaluated on the dataset of the DCASE2019 challenge. This dataset is composed of simulated data, convolution with measured room Impulse Responses (IRs). These models have not been evaluated on real data.
- Most architectures of the proposed models are based on prior knowledge, such as a maximum of two simultaneous sources, discrete DOAs at 10° intervals, etc. Moreover, the detected sound event and the position of this event in the room are sometimes estimated independently. When there are several events, it is therefore impossible to know which event is associated with which position.

Chapter 6

Sound event localization and classification on SECL-UMONS: Baseline model

Contents

This chapter is based on the following publication:

- Mathilde Brousmiche, Jean Rouat, Stéphane Dupont. "SECL-UMons Database for Sound Event Classification and Localization". In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

In this chapter, we present a more detailed description of the DCASE2019 challenge baseline (SELDnet), the only SELD model available when the new dataset was released. The baseline is used to generate benchmark scores for SECL-UMONS, the new dataset presented in Chapter 4. We slightly modify the model to introduce a benchmark score for real-time classification and localization.

6.1 Sound event localization and classification on SECL-UMONS

6.1.1 Baseline model

As a benchmark method, we employ the SELDnet model [101]. It was selected as the baseline for the DCASE2019 Challenge. The meta-parameters have the default values from the source code. SELDnet (Figure 6.1) is composed of several parts: the feature extraction, the feature process with Convolutional Recurrent Neural Network (CRNN), and the output estimation (event classification and position estimation).

Feature extraction The spectrogram is extracted for each of the 7 microphone channels using 512 points Discrete Fourier Transform with a Hamming window of 50% overlap. Only the 256 positive frequencies without the zeroth bin are kept. The phase and magnitude of the spectrogram are then extracted and used as separate features. The output of the feature extraction block is a feature sequence with an overall dimension of $T \times 256 \times 2 * 7$.

Feature process The model is a CRNN. The sequence of T spectrogram frames is fed to 3 convolutional layers with 64 filters of 3x3 kernel. The activation functions are the Rectified Linear Unit (ReLU). Max-pooling is applied only along the frequency axis. The temporal axis remains untouched to keep the resolution of the output unchanged from the input dimension. The temporal structure of the sound events is modeled using two bidirectional GRUs with 128 hidden units each.

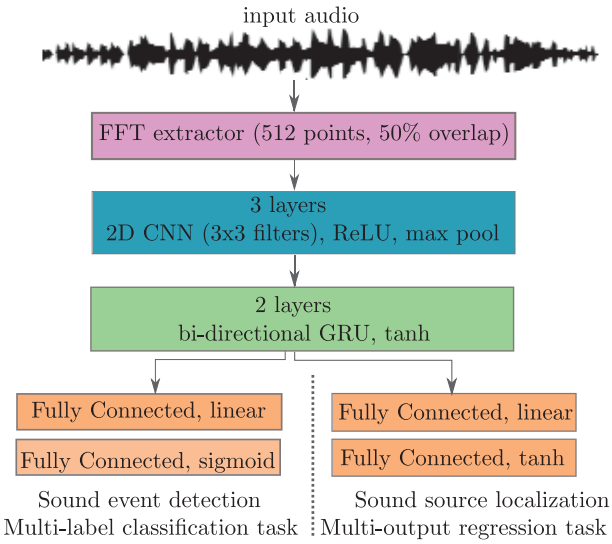


Figure 6.1. Benchmark model used to evaluate the new dataset.

Event classification The output of the recurrent layer is shared between two branches. The first branch is a Fully Connected (FC) layer with a sigmoid activation function used to classify each frame. For unilabel sequences, the class associated to the sequence is the class with the maximum output. For multilabel sequences, the classes associated with the sequence are the classes with an output higher than a threshold of 0.5.

Position estimation Finally, the second branch is a FC layer with a tanh activation function used to estimate the position of the event as a regression problem. More precisely, the output is multi-class regression. Therefore, the coordinates are estimated for each class and only the coordinates of the detected class are kept.

The network is trained with the combination of two losses: binary cross-entropy for the classification subtask and the Mean Squared Error (MSE) for the localization subtask. Adam optimizer is used with an initial learning rate of 0.001. The model is trained during 1000 iterations with early stopping,

when the global metric based on validation split has not decreased for 100 iterations, the training is halted. The metrics are explained in Section 6.1.2. Details of the network parameters can be found in Appendix B.1.

As the baseline is composed of bidirectional recurrent layers, real-time classification and localization are not feasible. We propose to modify the model to obtain preliminary results for potential real-time classifications & localizations. The bidirectional layers are replace by unidirectional recurrent layers.

6.1.2 Evaluation metrics

The baseline is evaluated using individual metrics for classification and localization estimations. A global metric is created to control the training early stopping.

Classification evaluation The standard polyphonic SED metrics are used: F-score and Error Rate (ER). The F-score is computed as:

$$F = \frac{2 \cdot \sum_{n=1}^{N} TP(n)}{2 \cdot \sum_{n=1}^{N} TP(n) + \sum_{n=1}^{N} FP(n) + \sum_{n=1}^{N} FN(n)} \tag{6.1}$$

where TP is the number of true positives, FP, the number of false positives, FN, the number of false negatives and N, the number of sequences.

The ER is computed as:

$$ER = \frac{\sum_{n=1}^{N} S(n) + \sum_{n=1}^{N} D(n) + \sum_{n=1}^{N} I(n)}{\sum_{n=1}^{N} E(n)} \tag{6.2}$$

where E is the total number of event classes in the sequence, S, D and I are the number of substitution, deletions and insertions respectively:

$$S(n) = \min(FN(n), FP(n)) \tag{6.3}$$
$$D(n) = \max(0, FN(n) - FP(n)) \tag{6.4}$$
$$I(n) = \max(0, FP(n) - FN(n)) \tag{6.5}$$

The ideal model will have an F-score of one and an ER of zero. In Section 6.2, the F-score is reported as percentages. We used the ER only in the global metric to control early stopping during training but not to evaluate the model during testing. The ER has no impact on the global metric for unilabel sequences because only one class per frame can be estimated. However, for multilabel sequences, the model has no constrain on the number of events estimated for each frame.

Localization evaluation The estimated location (x^E, y^E) and the ground truth location (x^G, y^G) from each frame are used to compute the estimated azimuth α^E and the ground truth azimuth α^G. The DOA error is computed as the absolute difference between the two angles:

$$DOAerror = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{K}\sum_{k=1}^{K}|\alpha_{nk}^{G} - \alpha_{nk}^{E}|\right) \tag{6.6}$$

$$\alpha = \arctan\left(\frac{y}{x}\right) \tag{6.7}$$

where K is the number of frames and N, the number of sequences.

In order to account where the number of estimated and groundtruth DOAs are unequal, the frame recall is used:

$$frame\_recall = \frac{TP}{TP + FN} \tag{6.8}$$

The ideal DOA error and frame recall are zero and one, respectively.

Combined SELD score Additionally, during the training, a combined SELD score is computed to perform early stopping:

$$SELDscore = \frac{SEDscore + DOAscore}{2} \tag{6.9}$$

$$SEDscore = \frac{ER + (1 - F)}{2} \tag{6.10}$$

$$DOAscore = \frac{\frac{DOAerror}{180} + (1 - frame\_recall)}{2} \tag{6.11}$$

The ideal SELD score is zero.

6.2 Results

As unilabel and multilabel sequences have different lengths, they are trained and evaluated separately.

During the training phase, as the onset and offset are not known, the ground truth classes and location of the events are assigned to each frame of the sequence. During the testing phase, for the unilabel sequences, only one class is selected for each sequence. The class is chosen by counting the most present class in all the frames. For multilabel sequences, the class is present in the sequence if the class output is greater than a threshold of 0.5 for at least one frame.

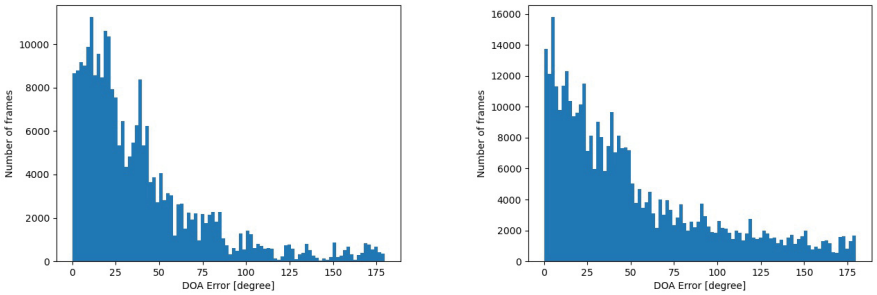
		F-score [%]	DOA error [degrees]
Unilabel	Bidirectional	96.04	38.83
	Unidirectional	89.26	41.77
Multilabel	Bidirectional	90.01	56.48
	Unidirectional	81.53	84.76

Table 6.1. Results of the classification and localization subtasks.

Table 6.1 presents the classification and localization results for the unilabel and multilabel sequences. The results are reasonably good. For comparison, some research has reported that humans have an average error of 11.75 degrees with a variation between 2 and 20 degrees. The use of unidirectional recurrent layers decreases the performances, probably because the first frames do not include information as long as the event does not occur.

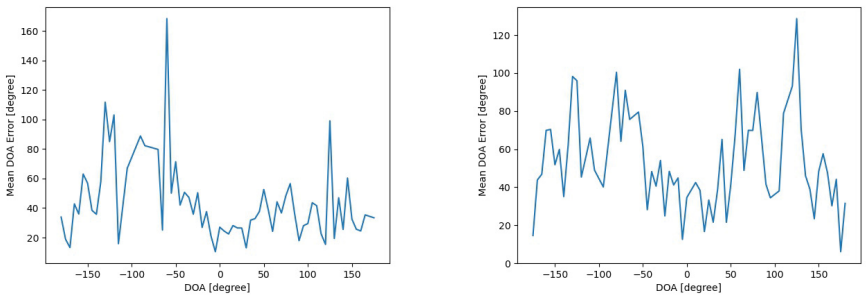
Figure 6.2 shows the DOA error histogram for unilabel and multilabel sequences. Most of the errors are small, only some sequences have errors greater

than 45 degrees. However, large errors occur more regularly for multilabel sequences. Figure 6.3 shows the DOA error depending on DOA for unilabel and multilabel sequences. The error varies significantly between DOAs, some DOAs are easier to locate than others. However, there can be a large variation between two close DOAs. Therefore, the localization difficulty is not specific to a particular area of the room.



(a) Unilabel sequences **(b)** Multilabel sequences

Figure 6.2. DOA error histogram for unilabel and multilabel sequences.



(a) Unilabel sequences **(b)** Multilabel sequences

Figure 6.3. DOA error depending on DOA for unilabel and multilabel sequences.

However, we notice a significant decrease in the classification and localization scores of multilabel sequences compared to the unilabel sequences. Moreover,

the network, as it was built in [101], cannot take into account a subset of our multilabel sequences, the sequences with two sounds from the same class but at a different location in the room. In this case, one of the events is therefore not present in the ground truth and is probably one source of the performance decrease.

Different strategies could be considered to address this problem. For example, as there are always two events in the sequences, it would be possible to add an additional output for the localization of the second event. This strategy is based on prior knowledge about the data. A more general strategy could be derived from object detection models with region proposal. Different time regions that may comprise an event are proposed by a model and for each proposed region, the class and the localization of the event in the room are estimated.

6.3 Model analysis

Several evaluations have been conducted for a deeper analysis: the impact of the FFT window size, the number of the microphones used, the localization problem formulation and the generalization ability.

6.3.1 Impact of the FFT window size and the number of microphones used

New evaluations have been conducted with varying sizes of the FFT window (number of channels = 7) and channel numbers of the microphone (FFT window = 512) on the unilabel sequences (Table 6.2).

Discussion The number of microphone channels has a minor impact on classification performance, but it is important for localization. The use of multiple microphones adds redundancy to the data and slightly improves results. Of course, localization with a single microphone is not possible. Indeed, localization is based on the difference in time and/or intensity between several microphones. The use of 4 microphones is sufficient to locate events. Again,

	# of channels				FFT window size		
	1	4	7		256	512	1024
F-score [%]	93.75	95.00	96.04	F-score [%]	94.21	96.04	95.83
DOA error [degress]	87.66	37.90	38.83	DOA error [degress]	48.35	38.83	41.07

Table 6.2. Influence of the number of channels and the size of the FFT window on the F-score and the DOA error for the unilabel sequences.

the use of 7 microphones adds redundancy and slightly improves results. However, this difference is probably due to variation in training rather than real improvement.

Each frame individually comprises more information with a larger window, which is beneficial for the classification. A size of 512 is enough for the classification. The size of the FFT window is also a compromise between time and frequency resolution that impacts the localization performance.

6.3.2 Localization problem formulation

In the previous experimentation, the event is located in space by estimating the x and y coordinates for each class. However, the event localization in space can be expressed in different ways such as x,y coordinates or azimuth angle. For unilabel sequences, different scenarios are compared:

- Estimation of x,y coordinates with a regression, one estimation for each class (coord multiregr);
- Estimation of x,y coordinates with a regression, the same estimation for each class (coord regr);
- Estimation of azimuth with classification (360 classes), the same estimation for each class (azi class);
- Estimation of azimuth with regression, one estimation for each class (azi multiregr);

- Estimation of azimuth with regression, the same estimation for each class (azi regr);

	coord multiregr	coord regr	azi class	azi multiregr	azi regr
DOA error [degrees]	38.83	27.72	79.91	76.49	73.85

Table 6.3. DOA error comparison for different localization problem formulations for unilabel sequences.

Table 6.3 presents the different results according to the localization problem formulation. The estimation of azimuth with classification has the worst results. The azimuth and coordinates estimations with a regression are better with one estimator for all classes than with one estimator for each class. The event location in the room does not depend on the event class. By using a single estimator for all classes, there are more examples to train this estimator. Finally, the estimation of x,y coordinates has better results than the azimuth estimation. The x,y coordinates are advantageous over azimuth due to their continuity. Indeed, azimuth regression is between 0 and 359 degrees. There is a big gap between 0 and 359 degrees when learning regression, there is actually only a difference of 1 degree.

Given the unilabel sequence results, only scenarios using x,y coordinates are compared for mutilabel sequences:

- Estimation of x,y coordinates with a regression, one estimation for each class, the same target vector includes the coordinate of the two events (coord multiregr);
- Estimation of x,y coordinates with a regression, the same estimation for each class, the models is composed of different localization outputs, one for each event (coord regr).

Each problem formulation has its advantages and disadvantages. With one estimator per class, there is a clear link between the estimated class and its location in the room. However, it is not possible to have two separate events with the same class but at different locations. On the other hand, with one

	coord multiregr	coord regr
DOA error [degrees]	56.44	55.74

Table 6.4. DOA error comparison for different localization problem formulations for multilabel sequences.

estimator par event, there is not a clear link between the estimated event and its location in the room. Indeed, we do not know which is event 1 and which is event 2. In our case, the DOA error is computed by making the best association between estimation and ground truth.

Discussion Estimating x,y coordinates could be more complicated than estimating the azimuth angle. Indeed, the model has to take into account the dependence between x and y axis. However, better performance is obtained with spatial coordinates than spherical coordinates due to the discontinuity of spherical coordinates. Since the Direction of Arrival (DOA) does not depend on the class, implementing a single regressor for all classes gives better results. However, this technique is only suitable for unilabel sequences. When several events are present simultaneously, one output must be created for each event. This can only be applied because the number of events is known. One issue remains, we don't know which event is associated with which x,y coordinates in the room.

New techniques could be proposed to solve this issue. For example, instead of analyzing the acoustic scene one frame at a time, it is possible to observe the acoustic scene as a whole and focus on the time regions of interest to detect acoustic events. Based on Faster RCNN [149] for object detection, the model would propose several time regions of interest that may comprise an event. Then, for each region of interest, the corresponding features would be extracted to classify the event, refine the time region and localize the event in the room.

6.3.3 The generalization ability

Moreover, following the split of data into test and training sets, it is possible to evaluate different aspects of the models such as the generalization ability (Section 4.5). Indeed, for unilabel sequences, instead of choosing randomly the data used in the test set, one subclass of each class is used as test data. For multilabel sequences, we evaluate the capacity of classification and localization for a duo of classes who have never been associated with each other during training. We notice that the classification and localization performances decrease (Table 6.5).

		split 1	split 2
Unilabel	F-score [%]	96.04	86.14
	DOA error [degree]	38.83	32.44
Multilabel	F-score [%]	90.01	66.15
	DOA error [degree]	56.44	88.82

Table 6.5. Comparison of performances between random split of data (split 1) and generalization ability (split 2) for unilabel and multilabel sequences.

Discussion The impact is greater for multilabel sequences than for unilabel sequences. For unilabel sequences, the variation between subclasses is small enough to be able to recognize and localize a subclass never seen during training. For the multilabel sequences, the model is perfected in the recognition of event duos rather than each event individually. For both sequences, the event localization is directly linked to the estimated class with the multi-regression. The estimation of a wrong class negatively affects the event localization.

6.4 In brief

Summary of Chapter 6

- In this chapter, we presented the baseline of DCASE2019 Challenge, named SELDnet. The model is composed of a Convolutional Recurrent Neural Network (CRNN) and two outputs. The first output estimates the event class and the second estimates the localization with regression.
- We evaluated the SECL-UMONS dataset with SELDnet for the unilabel and multilabel sequences.
- We also analyzed the impact on the performance of different parameters such as the FFT window size, the number of channels, the localization problem formulation and the train-test split distribution.

Perspective for Chapter 6

- Depending on localization problem formulation, different difficulties arise: the inability to encode two events located in different places but having the same class or two separate events in the same location or the inability to know which event is associated with which position in the room. New strategies must be explored to solve this issue, for example, a model based on region proposal.
- The new dataset can be used to classify and localize events. Based only on sound information, the performance is already good but not perfect. However, the dataset also includes visual data, including information that can improve results. The joint use of modalities is not trivial and different avenues will be explored in the remainder of the thesis.

Part IV

Audio-visual fusion for event classification and localization

Chapter 7

Related Work

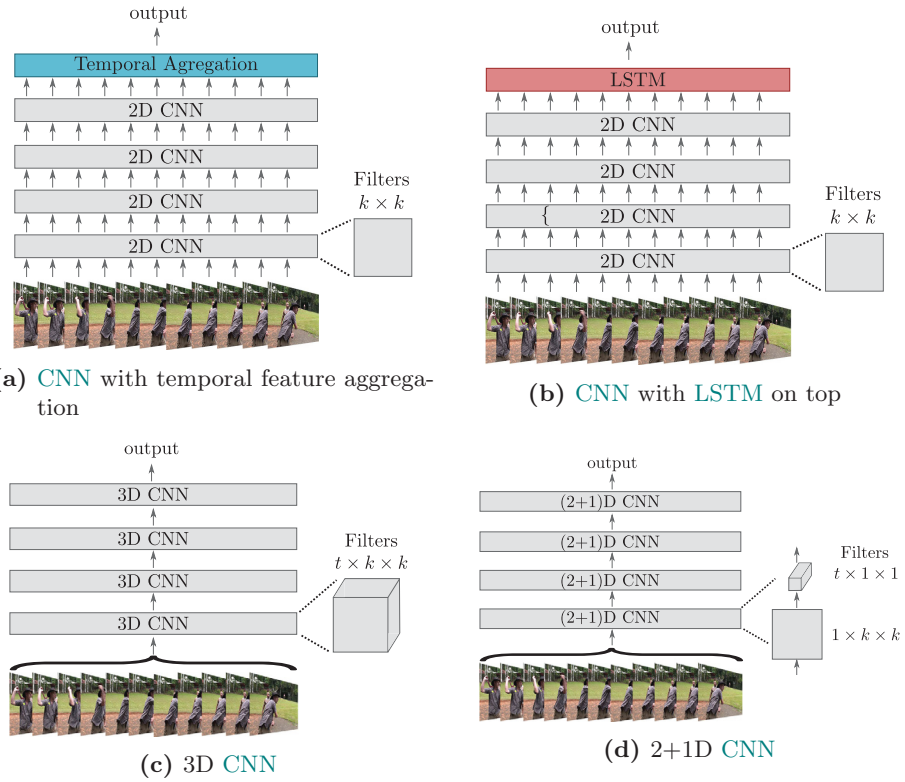
Contents

In this chapter, we present the related work for event classification and detection tasks. We start with systems based only on visual information, then continue with more recent works based on both audio and visual modalities. We finish by introducing the notion of inter-modality interaction, also called modality conditioning, and give several examples from the literature.

7.1 Visual event recognition

Inspired by the success of object classification within images [69,103,192–194], CNNs have also been applied in visual event recognition. Several methods have been proposed to take advantage of the temporal information. For example, the temporal feature aggregation was implemented with a temporal pooling layer [195–197].

RNNs are also effective to process temporal information. RNN on top of 2D convolutional layers was investigated in [195, 198–202] to take into account long-term dependencies. Li *et al.* went further and proposed a convolutional LSTM [203].



(a) CNN with temporal feature aggregation

(b) CNN with LSTM on top

(c) 3D CNN

(d) 2+1D CNN

Figure 7.1. Visual event recognition architectures. Video is decomposed into several frames. On one hand, each frame is processed by 2D CNN and the temporal information is aggregated with a temporal pooling layer (a) or a LSTM layer (b) to estimate a single output for the all sequence. On the other hand, all frames are processed by the convolutional layer composed either of a 3D kernel (c) or of the combination of two kernels (d).

Another approach was to extend 2D convolution kernels to 3D convolution kernels to learn spatio-temporal features [204–207]. As a number of very successful image classification architectures have been trained over the years, Carreira and Zisserman proposed to initialize 3D convolutional kernels with 2D kernels by "inflating" the kernel instead of using random weights [201]. On the other hand, to reduce complexity, the 3D convolution was decomposed

into two convolutions: a spatial 2D convolution and a temporal 1D convolution [208, 209].

At the same time, researchers tried to capture fine low-level motion by computing the optical flow [197, 208, 210, 211]. Most of the time, the optical flow is used conjointly with RGB information. Neural networks based on this technique have good results, but the optical flow is slow and computationally expensive. To tackle this problem, Fan *et al.* proposed a novel neural network designed to learn optical-flow like features in an end-to-end manner [212]. As the proposed network is end-to-end trainable, it can therefore be connected with a task-specific network to form a "deeper" end-to-end trainable architecture. Another strategy was proposed in [213] where the flow stream is used as a teacher for the RGB stream. However, the optical flow must be computed for training and the model remains computationally expensive during the training phase.

Finally, more recent techniques are composed of parallel paths that process different information such as static information and relations among frames [214] or spatial relations and temporal relations [215]. On the other hand, Feichtenhofer *et al.* proposed a model that involves a slow path, operating at low frame rate, to capture spatial semantics, and a fast path, operating at high frame rate, to capture motion at fine temporal resolution [216].

However, all these techniques do not exploit an important part of the video: the acoustic information.

7.2 Audio-visual event recognition

In recent years, only few works exploited the information present in the audio signal in the context of event/video classification. Most of the time, the audio signal is used in an additional stream to the visual stream (RGB and/or optical flow). However, no particular research was carried out to implement an efficient fusion of the two streams. The streams are fused at some point in the network through concatenation [84, 217–221] or at the decision level with the

average of the estimation of each stream [222, 223]. Some works went a little further by testing different levels of fusion in the network [224, 225].

As the multimodal network receives more information, it should match or outperform its unimodal counterpart, but this is not always the case. First, multimodal networks are often prone to overfitting due to their increased capacity. Second, different modalities overfit and generalize at different rates. Instead of investigating the fusion method, Wang *et al.* focused on training audio-visual data and proposed a complex Gradient-Blending training. They computed a loss that is an optimal blending of unimodal and multimodal loss based on the overfitting behaviors of the visual and audio information. Xiao *et al.* proposed a simpler strategy and randomly dropped the audio path during the training to take into account the differences in terms of "learning speed" of the audio and visual streams [226].

7.3 Audio-visual event detection

The release of the AVE dataset [70], at the end of 2018, has stimulated research in audio-visual event detection. Each work proposed different strategies to exploit the relevant information coming from the two modalities.

Tian *et al.*, the authors of the AVE database, defined the audio-visual event detection problem as the detection of events that are both audible and visible. They also proposed a baseline model (AVEL) for audio-visual event detection in the context of supervised and weakly-supervised learning [70]. The model is composed of visual and audio paths, fused with a Dual Multimodal Residual (DMR) fusion (Section 2.2.4). The network also includes an audio-guided visual attention mechanism to learn which visual region to look at based on the visual and audio information.

Lin *et al.* proposed to learn global and local event information in a sequence to sequence manner with the Audio-Visual sequence-to-sequence dual network (AVSDN) [227]. Global information is encoded with LSTM and Dual Multimodal Residual (DMR) fusion. Given both the fused global representations and local features of audio and video, the LSTM decoder generates the corresponding label segment by segment.

Wu *et al.* extracted the global representation of one modality and found the local segments in the other modality that are relevant to the event recognition and vice versa with the Dual Attention Matching (DAM) [228]. The task is divided into two subtasks: estimate the event category based on the overall sequences and differentiate background segments in the video.

In [229], the authors proposed two blocks: Audio-Visual Fusion Block (AVFB) and Segment-Wise Attention Block (SWAB). Audio-Visual Fusion Block is a fusion block based on Multimodal Factorized Bilinear Pooling (MFB) (Section 2.2.3) to generate spatial attention and a LSTM layer. The audio and visual features are projected to a common embedding space, concatenated and then fed to an LSTM for fusion. In addition to the spatial attention provided by Audio-Visual Fusion Block, Segment-Wise Attention Block highlights temporal segments of audio and visual paths independently. Indeed, all segments do not provide an equal amount of information about an event. The segment-level attention for one modality is based on global information of the same modality.

In [228] and [229], either the inter-modality interaction or the intra-modality interaction is explored. Ramaswamy explored both interactions conjointly with the Audio-Visual Interacting Network (AVIN) [68]. The classification is based on different information: high-level audio-visual associations and intra and inter-modality interactions. The high-level audio-visual association is computed with Multimodal Factorized Bilinear Pooling (MFB) Pooling. Because an event can occur only when both audio and visual content are synchronized at a particular instant, they used self and collaborative attention to capture intra and inter-modality interactions.

Finally, Xuan proposed a model similar to the baseline model with more interaction between visual and audio paths [230]. The cross-modal network is composed of an audio-guided spatial attention, as in the baseline, a self-attention module with global information, and a cross-modal adaptive co-attention module.

7.4 Modality conditioning

Before the release of the AVE dataset, most of audio-visual networks for event classification or detection were composed of a visual path on the one hand, and an audio path on the other hand. These two paths are fused with more or less complex mechanisms at some point in the network. Finally, the global multimodal feature is used to classify the event. However, very few works have tried to go further and create an interaction, also called conditioning, between both modalities.

Modality conditioning is the influence of a modality on another modality. It is the interaction between the paths of each modality inside the neural network. Interactions can be created by simple operations between paths such as an element-wise multiplication [223] or a sum [226] at different levels of the network.

More complex approaches to condition modalities have been explored, for example, the attention mechanism. Attention models were proposed in several applications such as object detection [48] or natural language processing with the self-attention mechanism [45]. Attention has been applied to video classification under the form of temporal and/or spatial attention [224, 231–235]. However, these models do not include an interaction between modality paths. With the release of the AVE dataset, new modality conditioning techniques were proposed. For example, audio conditions vision with a visual spatial attention guided by audio [70, 230]. Some works explicitly try to incorporate an inter-modality interaction layer with a collaborative attention [68, 230].

Another approach to condition one modality with the other is the Conditional Normalization (CN). Instead of focusing attention on a particular region of space or a particular temporal window, the CN highlights some feature maps based on a given input. Various forms of CN have proven to be highly effective across a number of domains and modalities: image stylization [236], speech recognition [237], visual question answering [238] and audio question answering [239].

7.5 In brief

Summary of Chapter 7

- In this chapter, we presented state-of-the-art strategies for event classification and detection. We started with strategies based only on visual information and then focused on audio-visual architectures.
- Inspired by the success of CNNs in image classification, most works use only visual information as images, optical flow or both together. They studied different strategies such as CNN with LSTM, 3D CNN, etc.
- Some works tried to add audio information with more or less complex fusions such as concatenation, summation, Dual Multimodal Residual (DMR) or Multimodal Factorized Bilinear Pooling (MFB).
- Finally, we introduced the notion of modality conditioning: the influence of one modality on the other modality. The conditioning can be implemented with different strategies such as simple operations, attention mechanisms or conditional normalization.

Perspective for Chapter 7

- This chapter mentions some works based only on visual information. On the other hand, it reviews all works on audio-visual event recognition to the author's knowledge. Even if there are more and more works on the subject, the number of works remains low in comparison with unimodal works. Moreover, most works have been published during the last 2 years.
- At the beginning of the thesis, the AVE dataset was not yet published. Very few audio-visual models for audio-visual event classification were present in the literature. Even if several fusion techniques were used, no comparison between them has been studied.

- Moreover, most of the modality conditioning techniques in the context of audio-visual event classification were investigated with the AVE dataset. Therefore, the modality conditioning between audio and visual paths has not been explored.

Chapter 8

Audio-visual event classification: fusion and conditioning

Contents

This chapter is based on the following publication:

- Mathilde Brousmiche, Jean Rouat, Stéphane Dupont. "Audio-Visual Fusion And Conditioning With Neural Networks For Event Recognition". In: *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2019.

In this chapter, we present the first experiments of fusion and conditioning. On one hand, we study different fusion techniques (concatenation, addition and Multimodal Compact Bilinear pooling (MCB)) in the context of audio-visual event classification. On the other hand, we introduce a technique of modality conditioning with the Feature-wise Linear Modulation (FiLM) layer, the information present in the audio modality is exploited to change the visual path behavior and vice versa. For these first experiments, we do not take into account the temporal information present in the visual modality and only use one image for the entire video. On the other hand, the entire soundtrack is exploited.

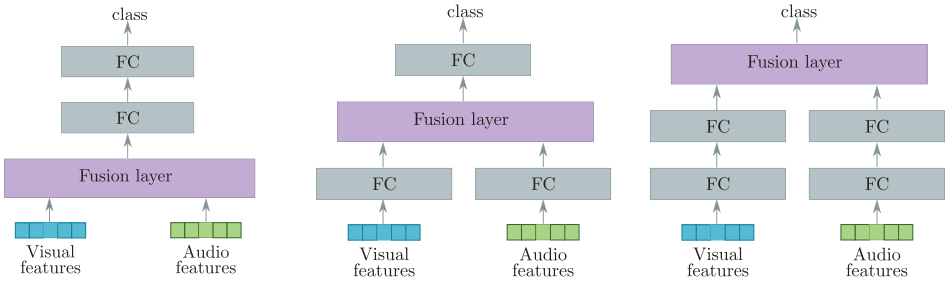
8.1 Methodology

For the first experiments of the thesis, the fusion and conditioning are studied separately.

8.1.1 Study of audio-visual fusion methods for event classification

We investigate three fusion methods to implement middle fusion at different levels of the architecture: concatenation, element-wise addition of the two feature vectors and Multimodal Compact Bilinear pooling (MCB) (Section 2.2.2).

Visual features and audio features are extracted, with pre-trained neural networks, from an image and the sound of the video. Then, the modality fusion can be made at 3 different levels in the architecture, the fusion technique depends on the level. Indeed, for the 1st fusion level (Figure 8.1a), as the visual and audio features have different shapes, we only used the concatenation. For the 2nd level (Figure 8.1b), as the visual and audio features pass through FC layers and have the same shape, less restrictive fusion methods can be applied (concatenation, addition and MCB). Finally, for the 3rd level (Figure 8.1c), the fusion block output has to be the class estimation, therefore only the addition can be applied. This level of fusion is not considered as a late fusion because the visual and audio networks are trained together with a single loss instead of being trained independently.



(a) 1st level **(b)** 2nd level **(c)** 3rd level

Figure 8.1. Fusion architectures for event recognition. Visual and audio features are obtained with DenseNet [194] and a CNN [240], respectively.

8.1.2 Modalities conditioning with FiLM

Perez et al. introduce in [241] a new kind of layer named Feature-wise Linear Modulation (FiLM). FiLM learns to adaptively impact the output of a neural network with an affine transformation to the network's intermediate features, based on a conditioning input. For example, this approach has previously been used in the framework of Visual Question Answering [241] and Acoustic Question Answering [239] problems. At first, features extracted from the textual question are used to modulate feature maps of images [241] or sounds [239].

In this thesis, we propose to study this approach to create interactions between image and sound instead of image (or sound) and text. One modality is used to "highlight" feature maps of the other modality with FiLM layers. We extract audio features and take these features as input to the FiLM layer to modulate feature maps of the visual path and vice versa. This approach couples and conditions the processing of the two modalities.

More formally, given an audio feature vector a of size D_a and visual features maps V of size $H_v \times W_v \times D_v$, FiLM learns functions f and h to compute γ_c and β_c as a function of input a:

$$\gamma_c = f_c(a) \qquad \beta_c = h_c(a) \tag{8.1}$$

γ_c and β_c modulate the activations $\mathbf{V}_c$, whose subscript refer to c_{th} feature map, via a feature-wise affine transformation:

$$FiLM(\mathbf{F}_c|\gamma_c, \beta_c) = \gamma_c \mathbf{V}_c + \beta_c \tag{8.2}$$

f and h can be arbitrary functions which are typically implemented with neural networks. FiLM blocks manipulate feature maps of a target, according to an input by scaling them up or down, negating them, shutting them off, selectively thresholding them (when followed by a ReLU), and more.

As in [241], the FiLM layer is combined to a Residual Block and the classification is made through a global average pooling followed by a fully connected layer (Figure 8.2).

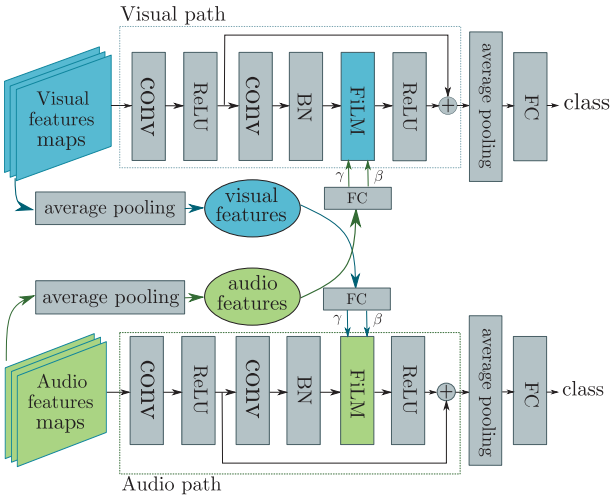


Figure 8.2. Our event classification model architecture with connections between visual and audio processing based on FiLM method. Visual and audio feature maps are obtained with DenseNet [194] and a CNN [240], respectively. γ and β parameters are computed by a FC layer. With the FiLM layer added in the residual block, the audio features extracted from the audio feature maps with average pooling are used to modulate visual feature maps and vice versa (modulation of audio feature maps with visual features).

8.2 Experimental details

8.2.1 Data description

For all experiments, we use a subset of the Kinetics dataset. Kinetics [242] comprises 400 actions carried out by humans. The clips last around 10 seconds. To facilitate the interpretation of the results, we select only 10 classes which have been chosen to be manifested both visually and aurally: blowing_nose, clapping, crying, finger_snapping, playing_drums, playing_guitar, sneezing, using_computer, whistling, yawning. For each class, we select 120 videos. The selection was made in order to have the same number of examples for each class, to ensure the presence of both modalities, and to ensure that video clips from the YouTube videos correspond to the selected class.

Given the small amount of data, visual and audio features are extracted with two neural networks pre-trained on ImageNet and AudioSet. Furthermore, data augmentation and 6-fold cross-validation have been used. The dataset is divided into three sets: the training set, which is composed of 80 examples per class, the validation set with 20 and the test set which also has 20 examples per class. Data augmentation is used for the training set: for each label, each image of this label is associated with each sound of the same label. The training set hence comprises 80(images)x80(sounds)x10(classes) = 64000 examples.

Visual Feature Extraction We used an ImageNet pre-trained deep learning model named DenseNet [194] to extract visual features from video clips. To do so, one frame, containing the event, is selected subjectively for each video. Then, 1920 dimensional feature vectors are extracted from each frame by taking the output of the global average pooling layer for the fusion method study and 7x7x1920 dimensional feature maps by taking the output of the last convolutional layer for the conditioning method.

Audio Feature Extraction We used a CNN on mel-band energies extracted from the sound [240] to extract audio features. This network is pre-trained with the AudioSet dataset. The entire sound sequence (from the whole duration of each video clip) is fed into the network. 1024 dimensional feature vectors are extracted by taking the output of the global average pooling layer

for the fusion method study, and 12x1x1024 dimensional feature maps by taking the output of the last convolutional layer for the conditioning method.

All networks are trained on 2 GPU (GTX1080 and Titan X) with the Tensorflow library. We used cross-entropy loss and the gradient descent optimizer with a learning rate of 0.001. The last activation layer is a softmax. Details of the network parameters can be found in Appendix B.2.

8.3 Results

8.3.1 Fusion method study

We first study the architectures described in Subsection 8.1.1.

Figure 8.3 reports the accuracy with either visual or audio modality alone and using the two modalities conjointly. Classification based on the audio modality only is better than the performance using only visual modality. It may be due to the fact that most of the classes are more distinguishable by sound than by image, for instance, crying, yawning, blowing nose.

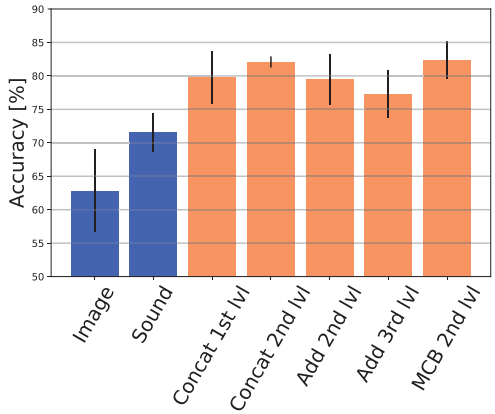


Figure 8.3. Comparison of unimodal (shown in dark blue) and multimodal (shown in light orange).

Table 8.1 presents the classification accuracy for each architecture. "Late fusion" is obtained by combining the output of the image and sound models when they are trained separately. Multimodal Compact Bilinear pooling (MCB) at the 2nd level has the best performance. Concatenation at the 2nd level gives comparable results with less variation between the different folds and has a shorter training time. On average, it takes 9 minutes to train the network composed of a concatenation at the 2nd level compared to 14 minutes for the MCB.

Fusion Strategies	Accuracy [%]
Late fusion	70.17 ± 5.53
Concatenation at 1st level	79.83 ± 3.96
Concatenation at 2nd level	$\mathbf{82.08} \pm 0.84$
Addition at 2nd level	79.50 ± 3.76
Addition at 3rd level	77.33 ± 3.58
MCB at 2nd level	$\mathbf{82.33} \pm 2.79$

Table 8.1. 6-fold cross-validation accuracy of different fusions.

Analysis by class Figure 8.4 presents the classification accuracy for each class for unimodal and multimodal models. Most classes are more easily classified with sound information except for *playing_guitar*, *playing_drums* and *using_computer*). These classes are the only classes that require the use of an object. Multimodal models are better than unimodal models for most classes but there is no one multimodal method that is best for all classes. For *crying* and *sneezing*, the visual information is not useful and does not bring any additional information to the sound. It decreases the performance of the model composed of an addition à 3rd level.

Figure 8.5 compares the confusion matrices of the model based only on visual information (Figure 8.5a), model based only on audio (Figure 8.5b) and the multimodal model composed of a concatenation at the 2nd level (Figure 8.5c).

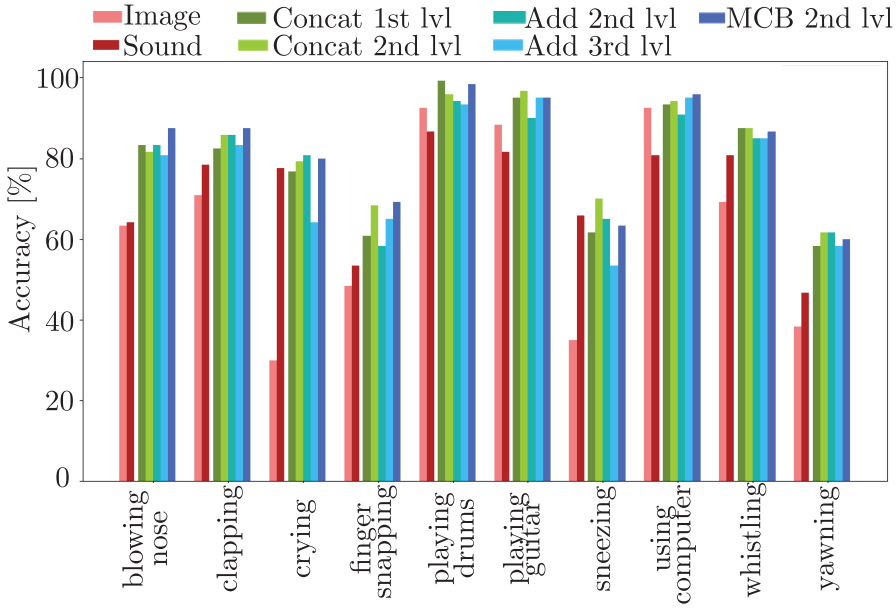
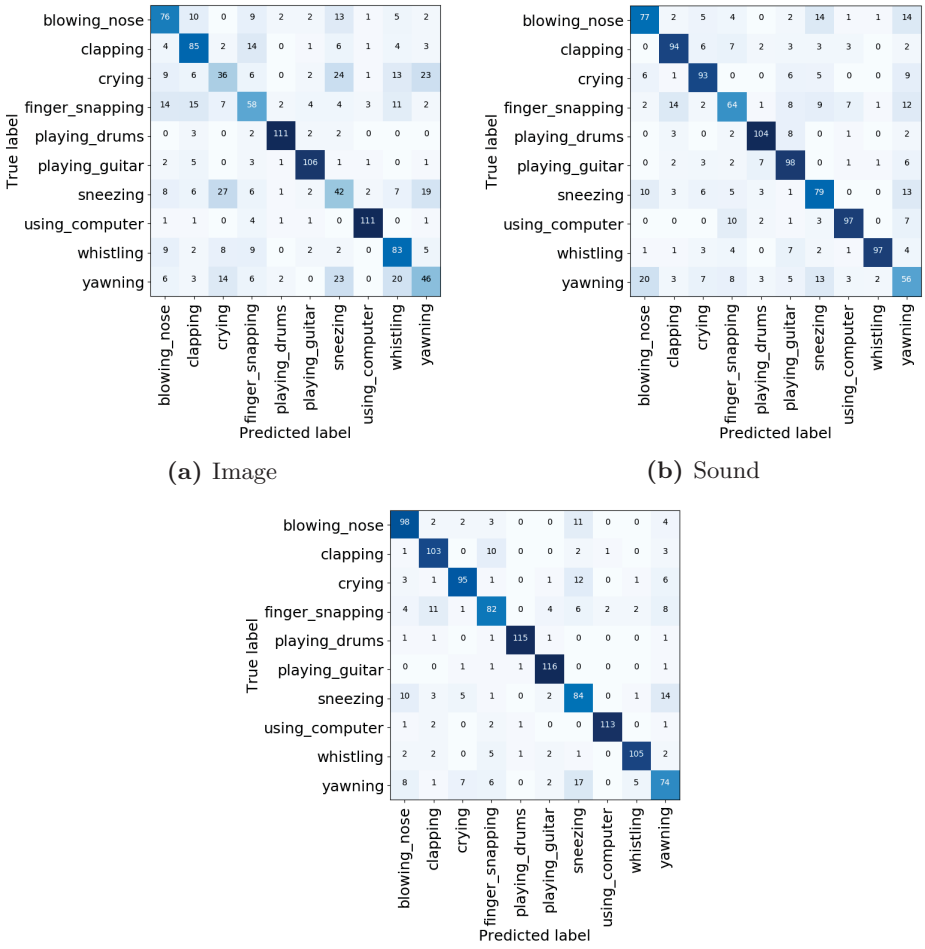


Figure 8.4. Accuracy per class for unimodal and multimodal models.

As a reminder, the visual information is extracted from a single image. There is, therefore, no notion of movement included in the visual features. We notice a confusion between several classes: *clapping*/*finger_snapping* (very similar classes) and *crying*/*sneezing*/*yawning*/*whistling* (classes mainly composed of video with a close-up on the person's face).

For the model based on sound information only, again different confusions occur: *sneezing*/*yawning*/*blowing_nose* (videos often include a baby doing the action with adult voices in the background), *finger_snapping*/*clapping* and *finger_snapping*/*using_computer* (produce similar sounds). The *yawning* class is the least well classified probably due to the absence of a particular sound in most of the videos.

Finally, several confusions also occur with the multimodal model. It is mainly the confusions common to both modalities (*blowing_nose*/*sneezing*/*yawning*, *finger_snapping*/*clapping* and *crying*/*sneezing*). However, they are less numerous.



(a) Image

(b) Sound

(c) Concat 2nd lvl

Figure 8.5. Confusion matrices for unimodal classification and the multimodal model composed of a concatenation at 2nd level.

8.3.2 Modality conditioning

In the conditioning architecture, described in subsection 8.1.2, visual modality is modulated by the audio modality and the audio modality is modulated by

the visual modality. The Feature-wise Linear Modulation (FiLM) layer is effective, as reported in Table 8.2.

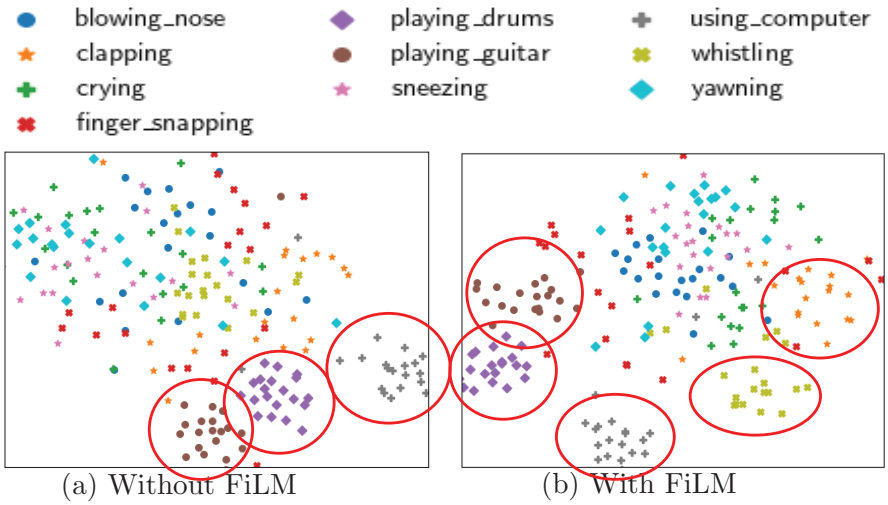
Accuracy [%]	Image	Sound
Without FiLM modulation	61.00 ± 5.11	66.67 ± 4.60
With FiLM modulation	75.75 ± 5.35	75.75 ± 3.14

Table 8.2. Performance of unimodal classification when adding a modulation from the other modality.

Figure 8.6 shows the Residual block output for image classification (average pooling output in Figure 8.2) with versus without the FiLM layer. t-distributed Stochastic Neighbor Embedding (t-SNE) [243] is applied to reduce the embedding dimension to 2D. We observe a better separation between classes with the FiLM layer, especially for clapping and whistling which are aurally-manifested classes. The spread of each cluster is measured by taking the average of the distances between the cluster center and each point of this cluster. The spread decreases from 0.329 without FiLM layer to 0.260 with the FiLM layer.

8.3.3 Impact of the presence of white noise

Three scenarios are tested: noise in the image only (Table 8.3), in the sound only (Table 8.4) and in both modalities (Table 8.5). Adding noise in the image has more impact on the performances than adding noise in the sound. The model composed of an addition at the 3rd level is the most affected method by the presence of noise, whether the noise is only in the image, in the sound or both. Models composed of MCB and concatenation at 2nd level remain the best techniques whatever the noise level except when the noise is present in both modalities, the concatenation outperforms MCB. Finally, models including FiLM layer have surprisingly good results (better than some of the multimodal models) when noise is added into both modalities compared to the performance when there is noise in images.



(a) Without FiLM (b) With FiLM

Figure 8.6. t-SNE visualization of the Residual Block output in the case of (a) image classification without FiLM layers and (b) image classification with FiLM layers.

8.3.4 Discussion

The fusions with concatenation or Multimodal Compact Bilinear pooling (MCB) have similar results. However, concatenation is easier to implement and requires less time for training. Concatenation is also the fusion technique least impacted by the presence of noise in both modalities simultaneously.

Both modalities do not always include relevant information for event recognition. In our case, the visual modality may even have a negative impact on the results of some classes. It would therefore be interesting to dynamically give more importance to one modality rather than the other. This attention would not be constant but computed dynamically for each video.

Relevant information to classify events is present in both modalities. We have shown that exploiting both audio and visual modalities through fusion or conditioning improves event recognition performance. However, when paired Student's t-test is applied to the fusion strategy results, we can not say that results are significantly different from each other. Experiments with more data should be done for more conclusive results.

	without noise	20 dB	15 dB	10 dB	5 dB
Image	62.83	55.25 (-12.06%)	48.83 (-22.28%)	37.67 (-40.05%)	24.25 (-61.40%)
Late fusion	70.17	63.00 (-10.22%)	56.50 (-19.48%)	45.67 (-34.91%)	33.42 (-52.37%)
Concat 1st lvl	19.83	76.33 (-4.38%)	70.83 (-11.27%)	63.00 (-21.08%)	54.25 (-32.04%)
Concat 2nd lvl	82.08	79.08 (-3.65%)	**76.08** (**-7.30%**)	70.92 (**-13.60%**)	**63.92** (**-22.12%**)
Add 2nd lvl	79.5	78.00 (**-1.89%**)	73.50 (-7.55%)	68.33 (-14.05%)	60.50 (-23.90%)
Add 3rd lvl	77.33	69.17 (-10.55%)	63.5 (-17.88%)	56.5 (-26.94%)	47.17 (-39.00%)
MCB 2nd lvl	**82.33**	**80.33** (-2.43%)	75.67 (-8.09%)	**69.00** (-16.19%)	60.00 (-27.12%)
Image with FiLM	75.75	60.17 (-20.57%)	57.08 (-24.65%)	60.58 (-20.03%)	55.08 (-27.29%)
Sound with FiLM	75.75	71.83 (-5.17%)	68.75 (-9.24%)	61.33 (-19.04%)	54.00 (-28.71%)

Table 8.3. Performances with different levels of white noise in image before the feature extraction. The number in brackets is the relative difference between the results with and without noise.

Although the conditioning technique improves classification performance, the conditioning method is not as efficient as fusion techniques. However, it shows promising results and other experiments must be done. The modality fusion and conditioning are complementary techniques allowing the simultaneous use of visual and audio information. It would therefore be interesting to combine the two techniques.

	without noise	20 dB	15 dB	10 dB	5 dB
Sound	71.58	67.83 (-5.24%)	66.92 (-6.51%)	64.75 (-9.54%)	60.75 (-15.13%)
Late fusion	10.17	69.83 (-0.48%)	69.25 (-1.31%)	68.83 (-1.91%)	67.92 (**-3.21%**)
Concat 1st lvl	19.83	79.08 (-0.94%)	78.67 (-1.45%)	78.80 (**-1.29%**)	76.08 (-4.70%)
Concat 2nd lvl	82.08	80.00 (-2.53%)	79.67 (-2.94%)	**79.50** (-3.14%)	**77.5** (-5.58%)
Add 2nd lvl	79.50	76.83 (-3.36%)	75.50 (-5.03%)	74.83 (-5.87%)	72.42 (-8.91%)
Add 3rd lvl	77.33	73.75 (-4.63%)	72.92 (-5.70%)	72.00 (-6.89%)	70.75 (-8.51%)
MCB 2nd lvl	**82.33**	**80.58** (-2.12%)	**79.75** (-3.13%)	78.75 (-4.35%)	76.83 (-6.68%)
Image with FiLM	75.75	75.75 (**-0%**)	75.42 (**-0.44%**)	74.75 (-1.32%)	72.42 (-4.39%)
Sound with FiLM	75.75	75.58 (-0.22%)	74.58 (-1.54%)	74.33 (-1.87%)	71.67 (-5.39%)

Table 8.4. Performances with different levels of white noise in sound before the feature extraction. The number in brackets is the relative difference between the results with and without noise.

	without noise	20 dB	15 dB	10 dB	5 dB
Late fusion	70.17	62.67 (-10.69%)	56.25 (-19.84%)	44.92 (-35.98%)	31.25 (-55.46%)
Concat 1st lvl	79.83	76.08 (-4.70%)	69.92 (-12.41%)	60.91 (-23.70%)	48.08 (-39.77%)
Concat 2nd lvl	82.08	**77.67** (-5.37%)	**74.42** (**-9.33%**)	**66.42** (**-19.08%**)	**55.17** (**-32.78%**)
Add 2nd lvl	79.5	75.83 (**-4.62%**)	69.42 (-12.68%)	62.17 (-21.80%)	50.67 (-36.26%)
Add 3rd lvl	77.33	65.5 (-15.30%)	60.00 (-12.68%)	50.00 (-21.80%)	36.5 (-36.26%)
MCB 2nd lvl	**82.33**	77.58 (-5.77%)	72.92 (-11.43%)	66.17 (-19.63%)	50.17 (-39.06%)
Image with FiLM	75.75	71.33 (-5.83%)	66.08 (-12.76%)	58.50 (-22.77%)	48.25 (-36.30%)
Sound with FiLM	75.75	71.58 (-5.50%)	67.00 (-11.55%)	60.67 (-19.91%)	48.75 (-35.64%)

Table 8.5. Performances with different levels of white noise in image and sound before the feature extraction. The number in brackets is the relative difference between the results with and without noise.

8.4 In brief

Summary of Chapter 8

- In this chapter, we studied several state-of-the-art fusion techniques (concatenation, addition and Multimodal Compact Bilinear pooling (MCB)). As expected, relevant information for event recognition exists both in visual and audio modalities.
- Inspired by perception principles of the brain, we also proposed to add Feature-wise Linear Modulation (FiLM) layers in the network to condition one modality with the other. More specifically, audio features give more importance to some visual feature maps and vice versa. The conditioning technique takes into account information coming from another modality and improves the classification performance.

Perspective for Chapter 8

- The current fusion techniques take into account all visual and audio information. Paying more attention to one modality than the other may be more similar to human behavior and a more effective technique.
- Separately, fusion and conditioning show both promising results for the joint use of visual and audio information. It may be promising to combine the techniques and create several interactions between modalities at different levels of the network.
- Only one frame per video is used to classify events. The proposed models (fusion and conditioning) do not take into account the temporal information present in the video. However, the notion of time is relevant for event recognition.
- Finally, the results were analyzed in detail thanks to the small size of the dataset. Unfortunately, the dataset is too small to draw defini-

tive conclusions. It is preferable to use several databases with more examples.

- These different points will be explored in the following chapter.

Chapter 9

Audio-visual event Classification: Multi-level fusion

Contents

This chapter is based on the following publication:

- Mathilde Brousmiche, Jean Rouat, Stéphane Dupont. "Multi-level Attention Fusion Network for Audio-visual Event Recognition". In: *Information Fusion*, 2020. [Submitted]

In Chapter 8, we introduced the notion of modality fusion and conditioning separately. In this chapter, we propose to jointly use both techniques. Furthermore, we introduce a more effective fusion technique based on an attention mechanism.

9.1 Multi-level Attention Fusion network

Inspired by the ability of living beings to pay attention to different regions, instants and modalities [1], we propose to compute a score for each modality and for each time window with the modality & temporal attention module. The attention module combines modality and temporal information to create a global feature that comprises the relevant multimodal and temporal information. We hence do not focus on spatial attention as done for instance in the image captioning task.

In addition to the fusion with the attention module, we propose to go further than modality fusion at a high level and include interactions between visual and audio paths with a Feature-wise Linear Modulation (FiLM) layer. In this section, we overview the Multi-level Attention Fusion network (MAFnet) and then detail the different components of the network.

9.1.1 Overview of the Multi-level Attention Fusion network

Figure 9.1 presents the architecture of MAFnet. As in [70], we split each video into T non-overlapping clips, where each clip is 1s long. We extract information for $K = 2$ modalities (visual and audio information). For each clip, we extract visual and audio feature maps with pre-trained CNNs. We have 2 input sequences: $\{F_1^1, \ldots, F_T^1\}, F_t^1 \in \mathbb{R}^{H_v \times W_v \times D_v}$ for the visual information and $\{F_1^2, \ldots, F_T^2\}, F_t^2 \in \mathbb{R}^{H_a \times W_a \times D_a}$ for the audio information. H, W and D are respectively the height, the width and the number of feature maps.

We reduce the feature maps with average pooling and feed the visual features ($\{x_1^1, \ldots, x_T^1\}, x_t^1 \in \mathbb{R}^{D_v}$) and audio features ($\{x_1^2, \ldots, x_T^2\}, x_t^2 \in \mathbb{R}^{D_a}$) in the modality & temporal attention module. This module is the combination of temporal and modality attentions. It is composed of self-attention and attempts to learn the attention scores λ_t^k with $t = 1, \ldots, T$ and $k = 1, \ldots, K$ to

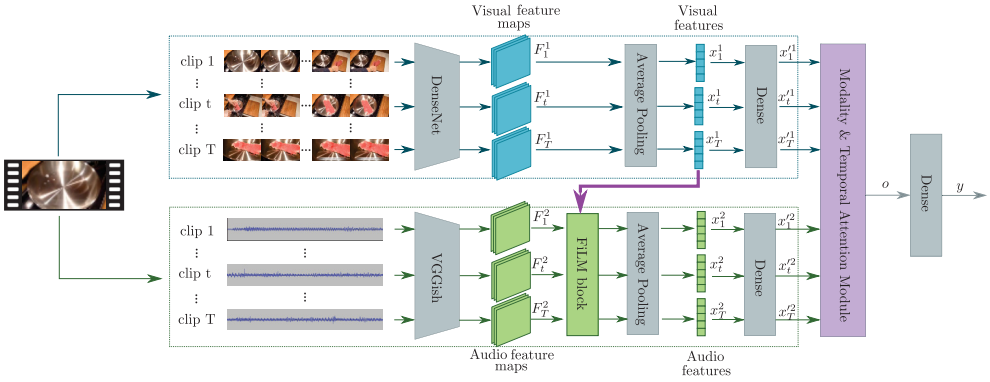


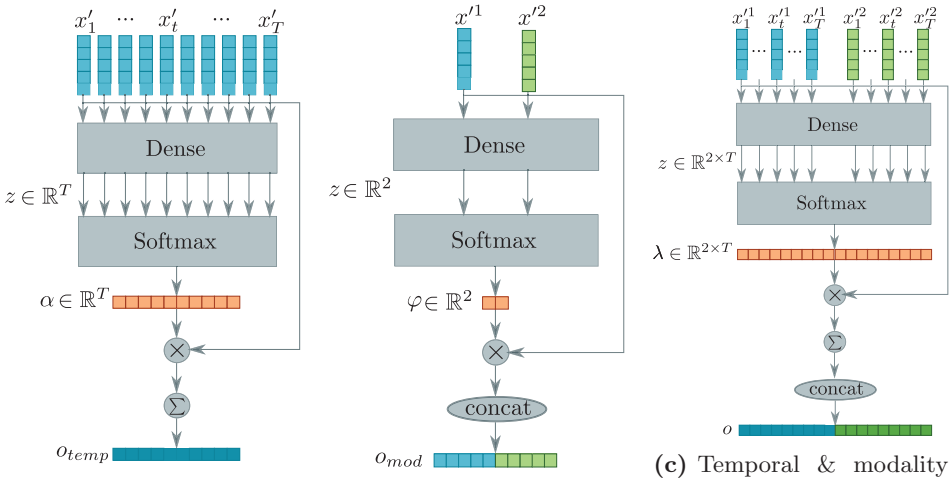
Figure 9.1. Multi-level Attention Fusion network (MAFnet): one video is split into T non-overlapping clips. Then, audio and visual information are extracted with two pretrained CNNs: DenseNet [194] for visual features and VGGish [244] for audio features. The clip features are further fed into the modality & temporal attention module to build a global feature comprising multimodal and temporal information. This global feature is then used to estimate the label of the video. A lateral connection between visual and audio paths is created trough the FiLM layer [241].

weight temporal and modality dimensions. We, therefore, obtain a temporal-multimodal representation of the entire video. The output of the network is $y \in \mathbb{R}^N$ with N the number of classes.

To go further than a simple fusion, we implement a lateral connection between visual and audio paths with the FiLM layer [241]. The visual modality influences the audio modality. Greater importance is given to some audio feature maps based on visual information. The FiLM layer is placed directly at the output of the audio feature extractor before reducing feature maps into vectors.

9.1.2 Temporal attention

The aim of temporal attention [224, 228] is to assign a positive weight score to each clip descriptors extracted from the video (Figure 9.2a). The score can be interpreted as the relative contribution of each clip to the recognition of the



(a) Temporal attention **(b)** Modality attention **(c)** Temporal & modality attention

Figure 9.2. Attention mechanisms. (a) Temporal attention: a score α is computed for each time window and the video-level feature representation o_{temp} with the sum. (b) Modality attention: a score φ is computed for each modality and the multimodal feature representation o_{mod} with the concatenation. (c) Temporal & modality attention: a score λ is computed for each time window AND modality and the global feature representation o with the combination of the sum over time windows and the concatenation over modalities.

target action, or the relative importance of each clip to generate an accurate global video representation.

Technically, for a given modality, given the input feature $X' = \{x'_1, \ldots, x'_T\}, x'_t \in \mathbb{R}^D$, the corresponding score $\alpha = \{\alpha_1, \ldots, \alpha_T\}$ over the T feature vectors is computed by

$$z_t = g_{att}(x'_t; \theta_{att}) = ReLU(W^T_{temp} x'_t + b) \tag{9.1}$$

$$\alpha_t = \frac{\exp(z_t)}{\sum_{j=1}^{T} \exp(z_j)} \tag{9.2}$$

where g_{att} is the temporal attention network parameterized by θ_{att}. g_{att} can take different forms such as a perceptron. z_t is an intermediate attention score, normalized with the softmax function.

We compute the video-level feature representation o_{temp} with the weighted sum of the clip features. The weights are the scores computed by the temporal attention module:

$$o_{temp} = \sum_{t=1}^{T} \alpha_t x'_t \tag{9.3}$$

9.1.3 Modality attention

In the context of speech recognition, Zhou *et al.* proposed a modality attention mechanism [245]. The attention mechanism fuses input from multiple modalities into a single representation by weighted summing the information from individual modalities. We propose to use a similar mechanism but use the concatenation of the weighted modalities instead of the sum (Figure 9.2b). In Subsection 9.2.5, we discuss the choice of modality fusion.

The attention module computes a score for each modality, the score is proportional to the contribution of the modality for the video classification.

Technically, at a given time, given the input feature $X' = \{x'^1, \ldots, x'^K\}, x^k \in \mathbb{R}^{D_k}$ with K the number of modalities, the score for each modality is computed by:

$$z^k = h_{att}(x'^k; \theta_{att}) = ReLU(W_{mod}^T x'^k + b) \tag{9.4}$$

$$\varphi^k = \frac{\exp(z^k)}{\sum_{j=1}^{K} \exp(z^j)} \tag{9.5}$$

where h_{att} is the attention network parameterized by θ_{att} and z^k is an intermediate attention score, normalized with the softmax function.

The multimodal feature o_{mod} is obtained by fusing the weighted unimodal features with a concatenation:

$$o_{mod} = concat([\varphi^1 x'^1, \ldots, \varphi^K x'^K]) \tag{9.6}$$

The modality attention module can dynamically choose the most relevant modality for a better classification of the events. Indeed, we can imagine that *Frying (food)* or *Truck* have strong visual information while *Violin* or *Flute* have strong audio information.

9.1.4 Modality & temporal attention module

We can combine the temporal and the modality attention modules to constitute the modality & temporal attention module (Figure 9.2c). The aim of the modality & temporal attention module is to assign a positive score for each modality and clip. Indeed, for example, in Figure 9.3, we notice that most of the time, the audio information is more relevant than the visual information except for the last clip where you can clearly see the food frying. This visual clip has the largest score and can be identified as the most relevant to classify the video as *Frying (food)*.

If we have the input:

$$X' = \begin{bmatrix} x'^1_1 & \cdots & x'^1_t & \cdots & x'^1_T \\ & \cdots & & \cdots & \\ x'^K_1 & \cdots & x'^K_t & \cdots & x'^K_T \end{bmatrix} \tag{9.7}$$

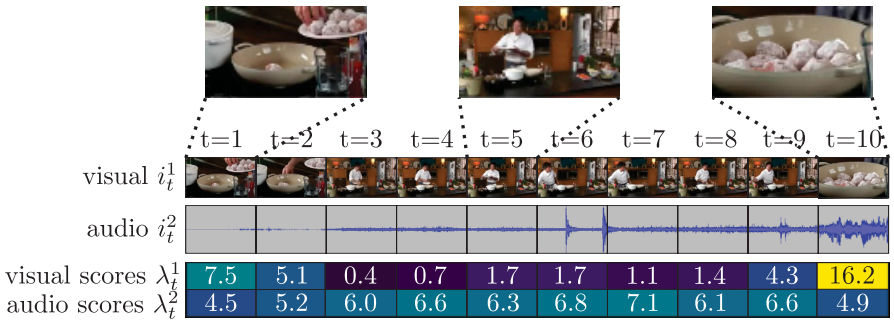


Figure 9.3. Visualization of the scores λ_t^k determined by the modality & temporal attention module for a video labeled *Frying (food)* of the AVE dataset. λ_t^k are in percentage due to the softmax and their sum is equal to 1. For t=1-2, the cook puts the food in the pan. For t=3-9, we hear the food frying and barely see it. At t=10, the cook starts talking but we have a clear vision of the food.

The equations of the attention module become:

$$z_t^k = f_{att}(x_t'^k; \theta_{att}) = ReLU(W_{mod+temp}^T x_t'^k + b) \tag{9.8}$$

$$\lambda_t^k = \frac{\exp(z_t^k)}{\sum_{j=1}^{T}\sum_{l=1}^{K}\exp(z_j^l)} \tag{9.9}$$

$$o = concat([\sum_{t=1}^{T}\lambda_t^1 x_t'^1, \ldots, \sum_{t=1}^{T}\lambda_t^K x_t'^K]) \tag{9.10}$$

The modality & temporal attention module is a self-attention mechanism. The score λ_t^k associated with each modality and time window vector $x_t'^k$ is computed based on the vector itself. It is the softmax (Equation 9.9) that normalizes the weights to each other. We add a dense layer in the path of each modality before the attention module because the attention module needs each modality to have the same dimension. Indeed, each vector $x_t'^k$ is processed by the same dense layer.

9.1.5 Lateral connection

We propose to go further than the "simple" fusion at high level by including a lateral connection to condition audio with vision earlier in the audio pathway. Indeed, most approaches do not exploit a possible interaction between the different paths. As presented in Section 8.1.2, the Feature-wise Linear Modulation (FiLM) layer can create a lateral connection between visual and audio paths. We use visual features as input to the FiLM layer to highlight feature maps of the audio modality (Figure 9.4).

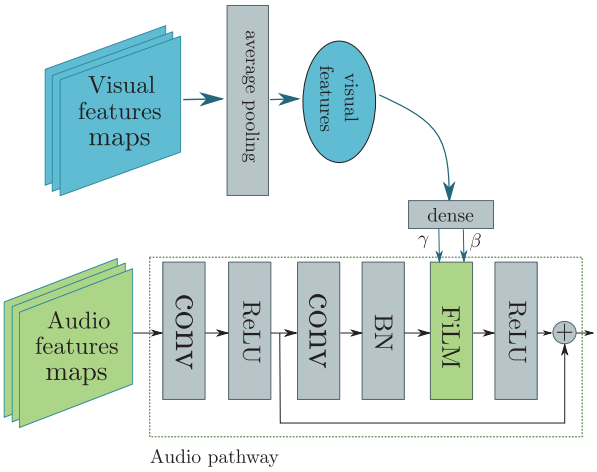


Figure 9.4. Lateral connection between visual and audio paths trough FiLM layer: The FiLM layer inside the residual block uses the visual features to modulate the audio feature maps. γ and β parameters are computed from a dense layer having its input from the visual features.

More formally, FiLM learns functions f and h to compute $\gamma_{t,c}$ and $\beta_{t,c}$ as a function of input x_t^1:

$$\gamma_{t,c} = f_c(x_t^1) \qquad \beta_{t,c} = h_c(x_t^1) \tag{9.11}$$

$\gamma_{t,c}$ and $\beta_{t,c}$ modulate the activations $\mathbf{F}_{t,c}^2$, whose subscripts refer to the t^{th} input and c^{th} audio feature map, via a feature-wise affine transformation:

$$FiLM(\mathbf{F}^2_{t,c}|\gamma_{t,c},\beta_{t,c}) = \gamma_{t,c}\mathbf{F}^2_{t,c} + \beta_{t,c} \tag{9.12}$$

f and h are implemented with dense layers.

9.1.6 Audio-visual training

Wang *et al.* noticed in [220] that multimodal networks are prone to overfitting due to their increased capacity. Moreover, different modalities overfit and generalize at different rates. So, they proposed a complex Gradient-Blending training. Xiao *et al.* noticed also different dynamics of training depending on the modality [226] and propose to randomly drop the audio path during the training. Unlike [226], when training unimodal networks, we notice that the audio network needs more epoch to reach overfitting compared to the visual network. Therefore, we follow the idea of [226] and randomly drop the weight update of the visual path, to train more the audio path.

9.2 Experimental results

9.2.1 Datasets

We evaluate our network on three public datasets: AVE [70], UCF51 [109] and Kinetics-Sounds [82].

AVE is a subset of AudioSet [91]. The dataset consists of 4143 videos from 28 event classes. Each video lasts 10 s. It covers a wide range of audio-visual events from different domains, e.g., human activities, animal activities, music performances, and vehicle sounds.

UCF51 is the second part of the UCF101 dataset [109]. Only the videos of the new 51 classes have sound information. UCF51 dataset consists of 6836 videos from 51 event classes. It concentrates on human actions. The mean video length is 7.0 s. The dataset is partitioned into three splits for training, validation and testing.

Kinetics-Sounds is a subset of the Kinetics dataset [201] and consists of only action classes that are potentially recognizable both visually and aurally. It consists of 21945 videos from 32 event categories. The mean video length is 9.7 s. In Chapter 8, we selected 10 classes of the Kinetics dataset ourselves while the Kinetics-Sounds dataset was proposed in different papers to test the performance of audio-visual models [217, 223, 226].

9.2.2 Feature extraction

Audio and visual features can easily be extracted from a new video using trained models [70, 224, 227, 228]. The extracted features are significantly smaller in size than the raw RGB frame and audio data. The networks performing the classification can hence be smaller

Visual feature extraction We use an ImageNet pre-trained deep learning model named DenseNet [194] to extract visual features from video. The video is split into T clips. As in [70], we choose T = 10, so each clip is one second long without overlapping. For each clip, we extract the output of the DenseNet last convolutional layer for 16 RGB video frames with a global average pooling over the 16 frames to generate one $7 \times 7 \times 1920$ dimensional feature map.

Audio feature extraction We use a VGG-like network [244] pre-trained on AudioSet to extract audio features. Again, the video is split into T=10 clips of one second each without overlapping. For each clip, we extract the output of the last convolutional layer of the network to generate one $12 \times 8 \times 512$ dimensional feature map.

9.2.3 Implementation details

The network is trained with cross-entropy loss and Adam optimizer with an initial learning rate of 0.001. Early stopping based on the validation accuracy is done, the training is halted when the validation accuracy has not improved since 50 epochs. During training, we randomly do not update the weights of the visual path. The model is implemented in Tensorflow [246]. The complete description of each network parameter can be found in Appendix B.3.

As UCF51 and Kinetics-Sounds datasets have different video lengths, feature vectors are zero padded to obtain the same length.

9.2.4 Event recognition performance

Table 9.1 presents event recognition results of MAFnet on AVE, UCF51 and Kinetics-Sounds datasets. We also compare our results with several state-of-the-art methods using different modalities, *i.e.* audio (A) and visual frames (V). For the UCF51 dataset, we report the average accuracy over three testing splits.

Briefly, I3D [201] is a CNN with 3D kernels. The network is initialized with kernels for image classification. R(2+1)D [208] is also a CNN. The 3D kernel is divided into 2 convolutions: 2D spatial convolution and 1D temporal convolution. SlowFast [216] includes two paths: a slow path, operating at low frame rate and a fast path, operating at high frame rate. MARS [213] uses the output of the optical flow network to train the RGB network. During the testing phase, only the RGB network is used. These networks do not have a separate feature extraction step, they can be trained end-to-end. For networks with a prior feature extraction step, Attention Cluster [218] is composed of several self-attention units on each modality. Then, modality features are concatenated to estimate the class. AVEL [70] includes a audio-guided visual attention. The modalities are fused with Dual Multimodal Residual (DMR).

MAFnet obtains the best accuracy performance on the AVE dataset among methods based on end-to-end training or feature extraction. End-to-end training methods have the advantage of being trained on larger datasets such as Sports1M or Kinetics to avoid overfitting and then the entire network is fine-tuned on smaller datasets. As the AVE dataset was built as an audio-visual set, the audio information is as important as the visual information. Therefore, as the end-to-end models take into account only the visual information, performances decrease. Furthermore, the AVE dataset includes classes from different events, unlike Sports1M and Kinetics datasets which include classes from human activities only. Models based on feature extraction have slightly better results than end-to-end training methods due to the use of audio information.

	model	inputs	pretrained dataset	Accuracy [%] AVE	UCF51	Kinetics-Sound
End-to-end training	I3D [201]	V	Kinetics	73.28	86.92	80.22
	R(2+1)D [208]	V	Sports1M + Kinetics	79.19	95.54	79.10
	SlowFast [216]	V	Kinetics	80.41	91.78	81.91
	MARS [213]	V	Kinetics	79.44	**97.83**	-
	model	**inputs**	**extraction network**	**AVE**	**UCF51**	**Kinetics-Sound**
Feature extraction	Attention Cluster [218]	V + A	DenseNet (V) + VGGish (A)	80.71	84.79	73.91
	AVEL [70] (our feat.)	V + A	DenseNet (V) + VGGish (A)	80.96	82.93	77.5
	AVEL [70] (their feat.)	V + A	VGG19(V) +VGGish-PCA(A)	85.02	81.04	-
	MAFnet(our)	V + A	DenseNet (V) + VGGish (A)	**90.86**	86.72	**83.94**

Table 9.1. Comparison with state-of-the-art models on AVE, UCF51 and Kinetics-Sound datasets. Each model was trained based on code available online. Models are split into two types: end-to-end training and feature extraction. End-to-end training models are trained on larger datasets and then fine-tuned on a smaller dataset. By contrast, feature extraction models are trained on feature previously extracted from the video. Depending on the model, input can be visual frame (V) and/or audio (A).

The UCF51 dataset comprises fewer classes with relevant audio information. Indeed, it includes classes that do not produce a particular sound signature or even video with irrelevant background noise. Our network is not as good as end-to-end training models which take advantage of pre-training on larger datasets and fine-tuning the entire network. On the other hand, models using feature extraction do not fine-tune extractor networks. However, MAFnet is the best model among the architectures that use audio-visual features extracted using models pre-trained on general purpose datasets such as ImageNet and AudioSet.

The Kinetics-Sound dataset as well as the UCF51 dataset are focused on human action, but comprises classes potentially recognizable both visually and aurally. Therefore, when the dataset comprises relevant audio and visual information, our network provides the best result. It is capable to take advantage of both modalities. Moreover, it has better integration of the audio and visual information than the other audio-visual models.

Figure 9.5 shows examples of output estimation from the different models. For examples a) to e), we observe that the background might impact the choice of the class. The raceway is classified as *Race car, auto racing* even if there is a bus (example a) or a motorcycle (example c). The field with a herd is classified as *Goat* but in this case, they are puppies (example e). Moreover, some specific elements in the video can fool models. The spoon and the plate (example b) may influence the I3D model in the choice of the *Frying (food)* class. In example d), visual models may be fooled by the round shape of the pan. In the case of examples c) and e), the audio modality is not distinctive enough to help the network.

We also note that some instruments can be difficult to distinguish (example f and g) or are occluded (example h). Some videos include several classes but are annotated with only one class (example i). Others are visually indistinctive (example j) but can be classified with the audio modality.

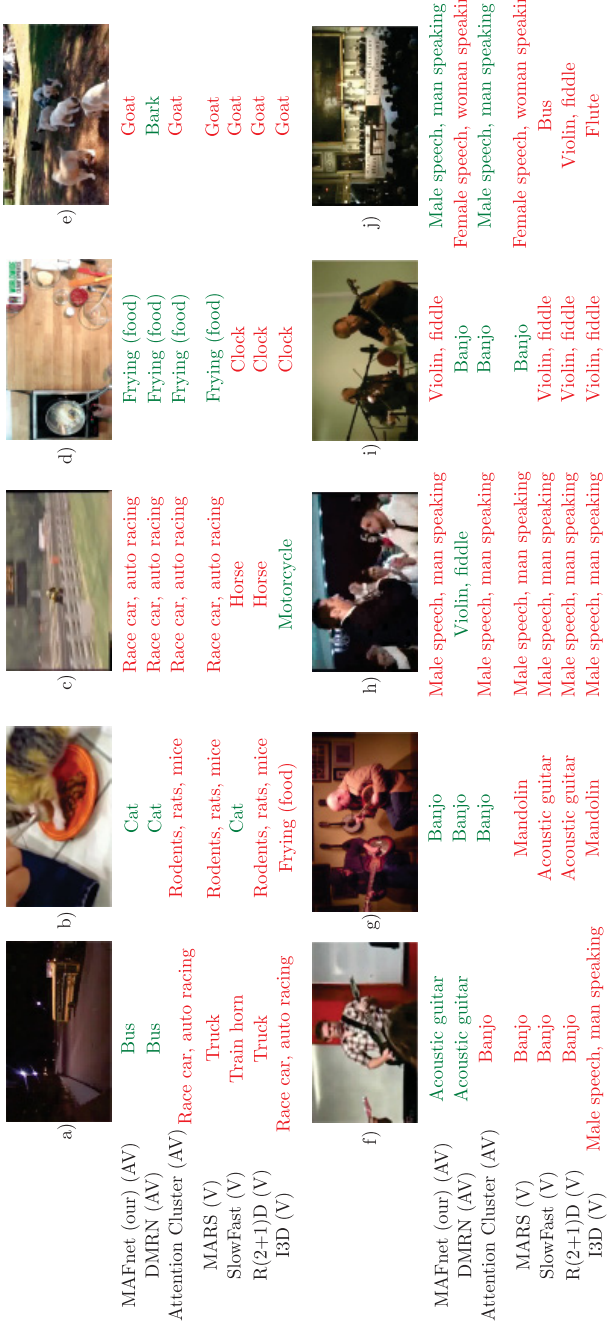


Figure 9.5. Output estimation of different visual only (V) models and audio-visual (AV) models for some example of the AVE dataset. Each model estimates one class per video. (Green: correct estimation, Red: False estimation)

9.2.5 Model analysis and discussion

In this section, we report studies to identify the impact of each module of the Multi-level Attention Fusion network (MAFnet). We work with the AVE dataset as the dataset assures the presence of the two modalities. We analyze the training method, the impact of the temporal attention, the modality attention and the combination of the two attentions, compare different fusion techniques and the impact of the modality conditioning.

Training method: Drop off Visual and audio paths do not have the same learning speed. Even without the additional convolutional layers comprised in Feature-wise Linear Modulation (FiLM), the training of the audio path needs more iterations than for the visual training. Inspired from [226], we investigate a new multimodal training technique by randomly dropping the update of the visual weights to allow the audio path to train longer. In Figure 9.6, we report the accuracy in function of the dropping rate of the weight update of the visual path. Dropping too often the visual path decreases results compared to training without dropping. We suppose that the visual path is not trained enough. It is also observed that not dropping enough the visual path gives also slightly poorer results. As the difference is quite small, it may be due to the performance variance after convergence.

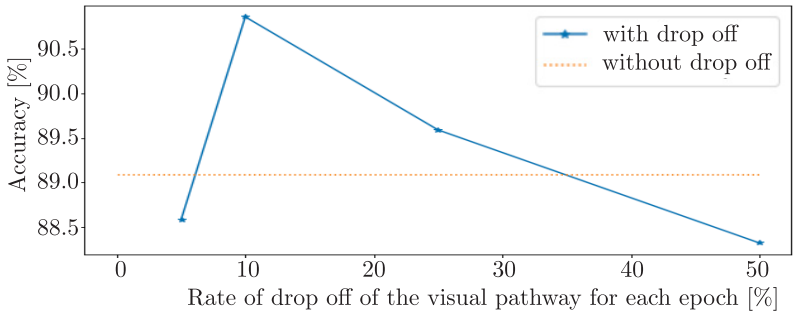


Figure 9.6. Accuracy of the event recognition of the AVE dataset when using different rates of dropping the weight update of the visual path during training.

Comparison of fusion techniques MAFnet creates a multimodal feature by concatenating the information coming from the visual and audio paths inside the Temporal & modality attention module (Figure 9.2c). We analyze the importance of using unimodal information versus multimodal information in the case of event recognition. We also test different fusion techniques present in the literature to determine the best fusion method: addition, concatenation, Multimodal Compact Bilinear pooling (MCB) [64] and the Dual Multimodal Residual (DMR) fusion [70].

As we want to test the fusion techniques, the experiments are made without the FiLM layer. In the case of the unimodal network, the network comprises only the temporal attention module without the modality attention module as only one modality is present.

fusion technique	Accuracy [%]
visual	75.63
audio	69.29
addition	84.77
concatenation	**89.34**
MCB	88.83
DMR	87.56

Table 9.2. Comparison unimodal versus multimodal event recognition and the use of different fusion techniques on AVE dataset.

From the results in Table 9.2, we notice that the dataset is easier to classify using visual information only than sound information only. Multimodal information increases performance compared to unimodal. Concatenation has the best result and is even slightly better than more complex fusion techniques like MCB or DMR.

Attention analysis We analyze the impact of each attention module. Table 9.3 presents the event recognition results without attention, with temporal attention only and with modality attention only. Again, the network does

not comprise the FiLM layer for this ablation study. The temporal attention allows taking into account the temporal context and dynamically highlights particular time windows. Not each time window comprises relevant information for the classification. The modality attention highlights a modality. Indeed, depending on the video a modality can have more contribution than the other. Each attention module has a positive impact on the accuracy and the combination of both attentions has the best result.

Attention type	Accuracy [%]
without attention	87.82
temporal attention	88.92
modality attention	88.66
modality & temporal attention	**89.34**

Table 9.3. Ablation study of the modality & temporal attention module on the AVE dataset.

Modality conditioning analysis We then analyze the impact of the lateral connection, the modality conditioning (Table 9.4). It is observed that adding FiLM in visual and audio paths provides better results than without any conditioning. However, conditioning only one modality is better than conditioning both modalities whatever the conditioned modality.

Figure 9.7 compares the embedding of the residual block just before and after the FiLM layer in the audio path (Figure 9.4). We use average pooling and t-SNE [243] to reduce the embedding dimension to 2D. We observe a better clustering of the different classes after including the visual information in the audio path.

FiLM location	Accuracy [%]
Add residual block without FiLM in both path	86.55
FiLM layer in both paths	87.62
FiLM layer in audio path	**90.86**
FiLM layer in visual path	90.61

Table 9.4. Evaluation of the lateral connection between visual and audio paths with FiLM layer on the AVE dataset.

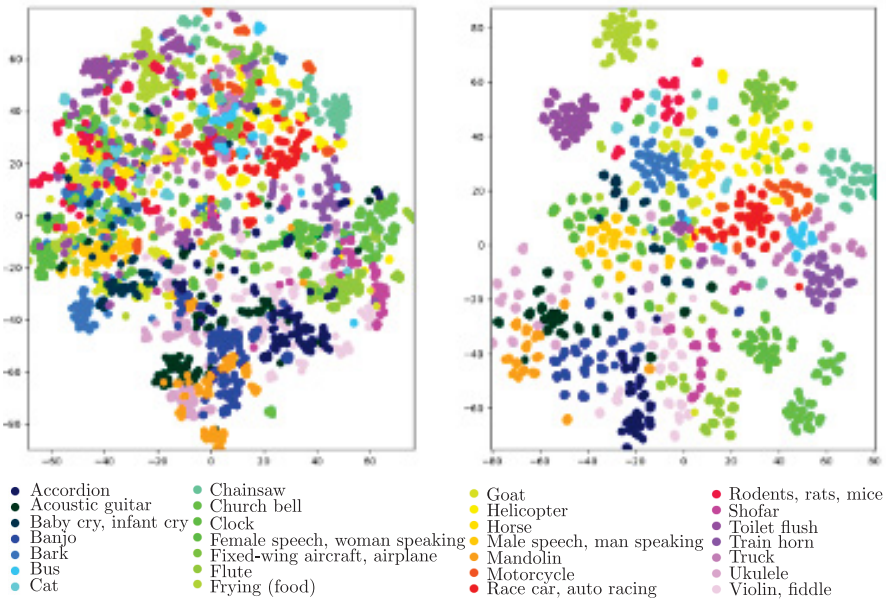


Figure 9.7. t-SNE visualization of the embedding of the residual block just before (left) and after (right) the FiLM layer in the audio path.

9.3 In brief

Summary of Chapter 9

- In this chapter, we present the Multi-level Attention Fusion network (MAFnet) for event recognition.
- Our network includes a modality & temporal attention module. It dynamically associates a score to each modality at each time window to highlight the relevant modality and time window.
- To go further than a 'simple' high-level fusion, we conditioned one modality with the other with a Feature-wise Linear Modulation (FiLM) layer. It highlights some audio feature maps based on visual data.
- Finally, to take into account the different learning dynamics of each modality, we randomly drop the weight update of the visual path.

Perspective for Chapter 9

- We notice the positive impact of the lateral connection (FiLM layer) in the multimodal model. FiLM is not the only technique to condition modality. It would be interesting to explore other techniques. With FiLM, the visual modality influences the audio modality independently in each time window. Another technique would be to influence each audio time window according to all visual time windows or vice versa.
- The current model is composed of a single connection (FiLM layer) and a fusion (modality & temporal attention module). Multiple connections between both modality paths should be tested to increase the interaction between modalities.
- In the modality & temporal attention module, the attention score associated with each modality and time vector is computed based on the vector itself. Taking into account more information simultaneously, such as information from both modalities, could be more effective.

Chapter 10

Event detection: Intra and inter-modality interaction

Contents

This chapter is based on the following publication:

- Mathilde Brousmiche, Stéphane Dupont, Jean Rouat. "Intra and Inter-Modality Interactions for Audio-Visual Event Detection". In: *The International Workshop on Human-centric Multimedia Analysis*, 2020.

In Chapter 9, the proposed model only classifies the event but does not estimate the beginning and end of the event (event detection). In this chapter, we present a new network for audio-visual classification and detection. The network models intra and inter-modality interactions and captures multimodal long-term dependencies. Finally, after evaluating the proposed model on the AVE dataset, we compare the new inter-modality interaction based on Multi-Head Attention (MHA) with the conditioning method based on Feature-wise Linear Modulation (FiLM).

10.1 Methodology

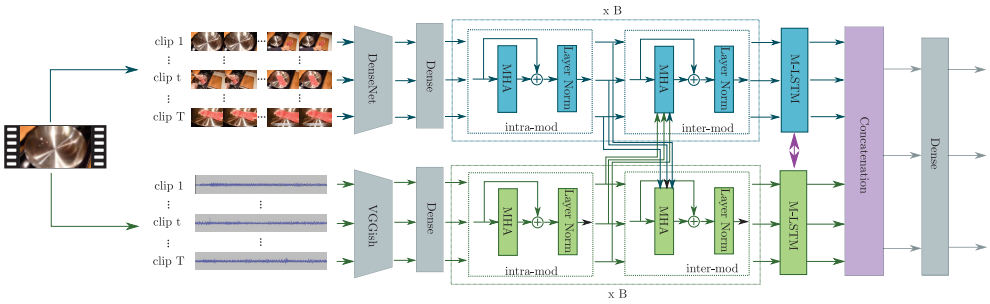


Figure 10.1. Our proposed model: one video is divided into T segments. Then, audio and visual information are extracted with two pretrained CNNs: DenseNet [194] for visual features and VGGish [244] for audio features. Each modality is further fed into B intra and inter-modality interaction blocks composed of MHA layers and a multimodal LSTM (M-LSTM). Finally, the two modalities are concatenated and the event class is estimated for each segment.

Our proposed network (Figure 10.1) is composed of different layers. As in [70], we split each video into T segments of length L. For each segment, we extract visual and audio features with pre-trained convolutional neural networks. So, we have 2 input sequences: $X^1 = \{x_1^1, \ldots, x_T^1\}, x_i^1 \in \mathbb{R}^{F_v}$ for the visual information and $X^2 = \{x_1^2, \ldots, x_T^2\}, x_i^2 \in \mathbb{R}^{F_a}$ for the audio information.

First, we model the intra as well as inter-modality interactions with several Multi-Head Attention (MHA) layers. The MHA creates a soft-alignment between the two modalities to facilitate the detection of audio-visual events. The

interaction of each temporal segment of one modality with each temporal segment of the other modality allows finding the segments of both modalities that include related information. This allows also finding the time segment where the event is simultaneously visible and audible. Then, the multimodal LSTM models the temporal information of the video for each modality but also captures multimodal temporal information. The temporal context of each time segment includes relevant information to recognize the event in that segment. The contextual information can be modality-specific but also multimodal. Finally, the two modalities are fused with a concatenation. The output of the network is $y \in \mathbb{R}^{N+1}$ with N the number of classes plus one background class. There is a non-background category only when audio and visual events are jointly observed.

In the rest of this section, we explain in detail the intra and inter-modality interaction blocks composed of Multi-Head Attention (MHA) layers and the multimodal LSTM.

10.1.1 Intra and inter-modality interactions

Multi-Head Attention (MHA) was introduced in the Transformer network [45] for automatic text translation. The attention mechanism learns the complex relationship between the source and target by aligning the source and the target. In our case, we want to learn the complex relationship between the modality and itself (intra-modality interaction) and between the two modalities (inter-modality interaction). The objective of intra-modality interaction is to compute attention scores that reflect the affinity of each time segment with each other within the same modality. The attention scores highlight the time segments that include related information and differentiate 'event' segments from 'background' segments. Inter-modality interaction is the same principle. Instead of computing the affinity between a modality and itself, the affinity is computed between the two modalities.

Scaled Dot-Product Attention An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted

sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key:

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{10.1}$$

where Q, K and V are respectively the query, the keys and the values. d_k is a scaling factor corresponding to the size of the keys. The operation QK^T results in a squared attention matrix containing the affinity between each row of the values V. In our case, it is the affinity between each time segment. In the case of the self-attention, the query, keys and values are the same input.

Intra-modality interaction: Single Modal Multi-Head Attention The Multi-Head Attention (MHA) is the idea of stacking several Scaled Dot-Product Attention attending the information from different subspace representations of the query, keys and values. The query Q, keys K and values V are projected into h subspaces through dense layers. Scaled Dot-Product Attentions are then applied in parallel on each projection. The h output values are then concatenated and projected again to obtain the final value (Figure 10.2):

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{10.2}$$

$$MultiHead(Q, K, V) = Concat([head_1, ..., head_h])W^O \tag{10.3}$$

where W_i^Q, W_i^K, W_i^V and W^O are the projection matrices.

By taking the same input for the query, keys and values, the QK^T matrix represents the intra-modality interaction and highlights time segments with similar content. As in the Transformer network [45], the MHA layer is followed by residual connection and a layer normalization (Figure 10.1).

Inter-modality interaction: Multimodal Multi-Head Attention In the single modality attention, the query, keys and values are the same input. Inspired by [247], to create an inter-modality interaction, the keys and the values are computed based on one modality and the query is computed based on the

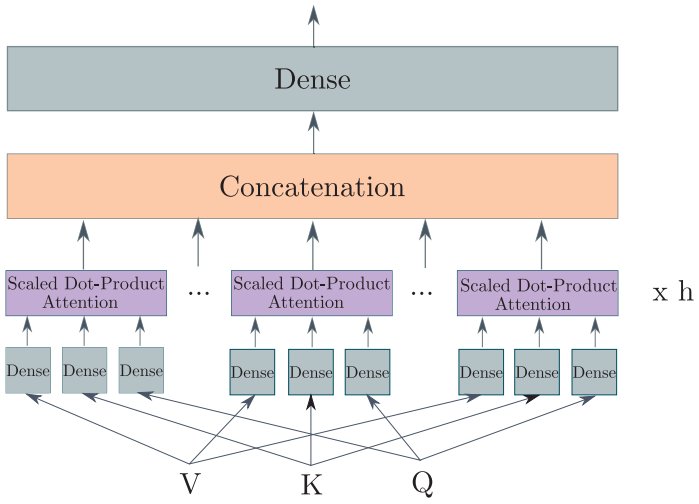


Figure 10.2. Multi-Head Attention (MHA) Layer. The query Q, keys K and values V are projected into h subspaces through dense layers. Scaled Dot-Product Attention is applied in each subspace. The outputs are then concatenated and projected again.

other modality. This time, the QK^T matrix represents the affinity between both modalities. The Multimodal MHA finds the segments where the event is simultaneously visible and audible. The inter-modality interaction block is composed of a multimodal MHA layer, a residual connection followed by a layer normalization (Figure 10.1).

10.1.2 Long Short-Term Memory

Single Modal LSTM LSTM networks, introduced in [38], can learn long-term dependencies. As a reminder, equations 10.4, 10.5, 10.6, 10.7 and 10.8 formally describe the memory input, the input gate, the forget gate, the output gate and the memory unit of a regular LSTM in the forward pass. Figure 10.3a shows a pictorial illustration of a regular LSTM unit.

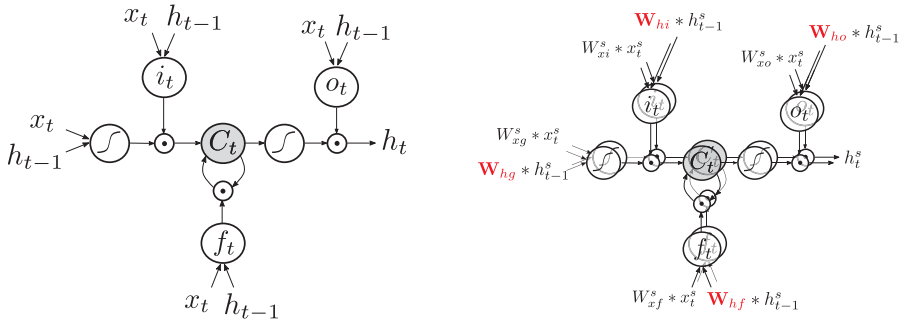
$$g_t = \varphi(W_{xg} * x_t + W_{hg} * h_{t-1} + b_g) \tag{10.4}$$
$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + b_i) \tag{10.5}$$
$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + b_f) \tag{10.6}$$
$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + b_o) \tag{10.7}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot g_t \tag{10.8}$$
$$h_t = o_t \odot \varphi(C_t) \tag{10.9}$$

The LSTM is able to model long-term dependencies in sequential data because C_t can selectively "remember" (store) or "forget" (erase) past information at each time step. Moreover, the LSTM can explicitly model temporal relationships over the entire sequence because the weights W are shared across time steps.

Multimodal LSTM Ren et al. developed a new multimodal LSTM which can explicitly model the long-term dependencies both within the same modality and across modalities [248]. The key idea is to selectively share weights across different modalities during the forward pass. Therefore, the model is composed of a visual LSTM and an audio LSTM, but some weights are shared between the two LSTMs. The modifications are illustrated in Figure 10.3b and formally expressed in the following equations:

$$g_t^s = \varphi(W_{xg}^s * x_t^s + \mathbf{W}_{hg} * h_{t-1}^s + b_g^s),\ s = 1\ to\ S \tag{10.10}$$
$$i_t^s = \sigma(W_{xi}^s * x_t^s + \mathbf{W}_{hi} * h_{t-1}^s + b_i^s),\ s = 1\ to\ S \tag{10.11}$$
$$f_t^s = \sigma(W_{xf}^s * x_t^s + \mathbf{W}_{hf} * h_{t-1}^s + b_f^s),\ s = 1\ to\ S \tag{10.12}$$
$$o_t^s = \sigma(W_{xo}^s * x_t^s + \mathbf{W}_{ho} * h_{t-1}^s + b_o^s),\ s = 1\ to\ S \tag{10.13}$$
$$C_t^s = f_t^s \odot C_{t-1}^s + i_t^s \odot g_t^s,\ s = 1\ to\ S \tag{10.14}$$
$$h_t^s = o_t^s \odot \varphi(C_t^s),\ s = 1\ to\ S \tag{10.15}$$

The superscript s indexes each modality. S is the total number of modalities in input data, in our case, $S = 2$. The weights with superscript s (e.g. $\mathbf{W}_{xg}^s$) are



(a) Memory Cell of Single-modal LSTM **(b)** Memory Cell of Multimodal LSTM

Figure 10.3. Comparison of single-modal LSTM and multimodal LSTM.

NOT shared across modalities, but only across time steps as in a conventional LSTM. The other weights without the superscript (e.g.$\mathbf{W}_{hg}$) are shared across both modalities and time steps. Another important property of the model is that the memory unit C is NOT shared among modalities.

10.1.3 Fully Supervised Learning for Event Detection

For each input audio-visual segment, the outputs of the two multimodal LSTMs are concatenated and fed to a FC layer with softmax function to estimate the probability distribution over $N+1$ event categories. For the supervised event detection task, the event label of each segment is known during training.

10.1.4 Weakly-Supervised Learning for Event Detection

For the weakly supervised event detection task, we have access to only a video-level event tag, and we still aim to estimate segment-level labels during testing (weakly supervised). Thus, the weakly supervised task is formulated as a Multiple Instance Learning (MIL) problem [249]. As in [70], the estimations for each segment are aggregated to obtain a video-level estimation using MIL pooling:

$$\widehat{y} = g(y_1, y_2, ..., y_T) = \frac{1}{T}\sum_{t=1}^{T} y_t \tag{10.16}$$

where y_1, ..,y_T are the estimations from the last FC layer of our network for each segment and $g(.)$ averages overall estimations. During testing, the event category is estimated for each segment.

10.2 Experiments and Results

10.2.1 Data description

AVE dataset [70] is a subset of AudioSet [91]. The dataset consists of 4143 videos from 28 event classes. Each video lasts 10 s. However, the duration of the events in these videos spans from a minimum of 2 seconds to a maximum of 10 seconds. Each video is divided into 10 one-second segments. A label is associated with each segment. The detection precision is therefore one second. The dataset covers a wide range of audio-visual events from different domains, for example, human activities, animal activities, music performances, and vehicle sounds.

10.2.2 Feature Extraction

Visual Feature Extraction We used an ImageNet pre-trained deep learning model named DenseNet [194] to extract visual features from the video. The video is divided into T segments. As in [70], we choose T = 10, so each segment is one second long without overlapping. For each segment, we extract the output of the DenseNet last convolutional layer for 16 RGB video frames and then use global average pooling over the 16 frames and the feature maps to generate one 1920 dimensional feature vector.

Audio Feature Extraction The audio features are extracted with VGG-like network [244] pre-trained on AudioSet [91]. Again, the video is divided into

T=10 segments of one second long without overlapping. For each segment, we extract the output of the last convolutional layer of the network and use global average pooling to generate one 512 dimensional feature vector.

10.2.3 Implementation details

The number of neurons in each layer except the dense output layer is 512. We have 2 intra and inter-modality blocks. We use cross-entropy loss and we train the model by using Adam optimizer with an initial learning rate of 0.001 during 50 epochs. We use Tensorflow [246] and Keras [250] libraries. Details of each network parameter can be found in Appendix B.4.

10.2.4 Event Detection Performance

Model	Fully-Supervised Accuracy	Weakly-Supervised Accuracy
Only visual	65.5	61.7
Only audio	64.1	59.2
AVEL [70]	72.7	66.7
AVSDN [227]	75.4	74.2
DAM [228]	74.5	-
AVIN [68]	75.2	69.4
AVFB + SWAB [229]	74.8	68.9
cross-modal net [230]	77.1	**75.7**
Proposed model	**77.8**	72.4

Table 10.1. Performance comparison of current state-of-the-art methods for fully-supervised and weakly-supervised event detection tasks.

The result of our proposed network is compared to recent models for both fully-supervised event detection and weakly-supervised event detection (Table

10.1). All models are described in the section 7.3. Our multimodal method outperforms a model using only visual or only audio information. Figure 10.4 shows the accuracy of a few individual event classes for the case of visual, audio and multimodal models. We notice that audio performs well for some event classes while visual performs favorably for other classes. However, multimodal has the best result for almost every class.

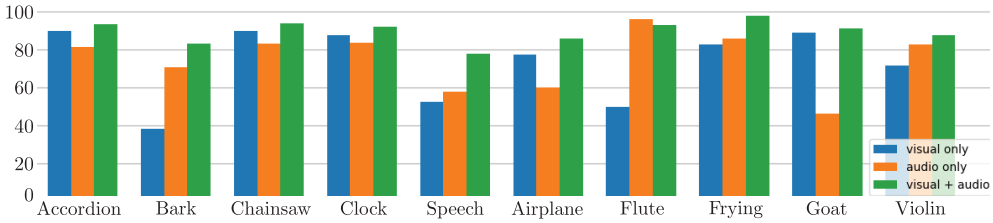


Figure 10.4. Accuracy of a few selected event categories obtained using only visual information, only audio information and our proposed model (visual + audio).

For the supervised event detection task, our proposed model obtains the best result among current state-of-the-art models. It has comparable results for weakly supervised event detection. We observe the contribution of including both intra and inter-modality interactions. Indeed, methods that exploit only inter-modality interaction (DAM) or only intra-modality interaction (AVFB+SWAB) have poorer results compared to methods that model both intra and inter-modality interactions (AVIN, cross-modal net and our proposed model). The accuracy of most architectures increases by 5.8 to 6.0 % when ground truth is available for each segment. Only the accuracy of AVSDN and cross-modal net increases only by 1.2 and 1.4 % respectively. They do not take full advantage of the information available at the segment level.

10.2.5 Model Analysis and Discussion

Ablation study The impact of each module is presented in Table 10.2. Each module performs well when executed separately, but the best result is obtained when the two modules are combined. Each module captures information that the other modules can not capture and are better together.

Model	Fully-Sup. Accuracy	Weakly-Sup. Accuracy
only intra + inter-mod	75.3	71.2
only M-LSTM	74.1	70.0
intra + inter-mod + M-LSTM	77.8	72.4

Table 10.2. Ablation study. The impact of each module is shown for the fully-supervised and weakly-supervised tasks.

LSTM analysis In this part, we analyze the impact of the weight sharing between the visual and audio LSTMs (Table 10.3). It is observed that the lack of information sharing between the two LSTMs decreases performance. When only one LSTM is used for both modalities (all weights are common between the visual and audio LSTMs), the performance slightly decreases for fully-supervised training and remains constant for weakly-supervised training. As visual and audio information come from the same source and have a strong coupling with the inter-modality blocks, they can benefit from weight sharing. Weight sharing, in addition to the inter-modality interaction, could force visual and audio features to converge towards a common representation when they represents the same element.

Model	Fully-Sup. Accuracy	Weakly-Sup. Accuracy
LSTM	76.07	72.1
Multimodal-LSTM	77.8	72.4
Unique LSTM	77.5	72.4

Table 10.3. LSTM analysis. Impact of the weight sharing in the LSTM layer for fully-supervised and weakly-supervised tasks.

Qualitative analysis The first type of errors is the event temporal detection. The event class is correct, but the estimation of the beginning and end of

the event is not accurate (Top in Figure 10.5). Another error is the confusion between similar classes, there is confusion between the instrument classes (Mandolin, Acoustic guitar, banjo, etc.) (middle in Figure 8.3) or between engine classes (Bus, Truck, Motorcycle, etc.). Finally, the model is occasionally fooled by some elements of the video and estimates a class unrelated to the ground truth (Bottom in Figure 8.3).

Sigmoid analysis As a reminder, the network estimates the probability distribution over $N+1$ classes, N the number of classes plus one background class. The class of the segment is determined by taking the maximum probability. However, in the case of the weakly supervised task, the background class is not explicitly represented during the training. Therefore, the background class is never chosen for a segment during testing which greatly impacts classification performance.

We propose to use sigmoid activation function instead of the softmax. The estimated class is either background if every output is smaller than a threshold of 0.5 or the class with the maximum output.

Accuracy [%]	softmax	sigmoid
Fully-supervised	77.8	71.74
Weakly-supervised	72.4	70.75

Table 10.4. Performance comparison of softmax and sigmoid activation function for fully-supervised and weakly-supervised event detection tasks.

The classification performance is not as good as with the softmax activation function. Indeed, the weakly supervised model can estimate 'background' class for any segment. However, there is much more confusion between classes as shown by the results of fully-supervised classification.

10.2.6 Conditioning comparison

Finally, we compare two modality conditioning methods presented in this thesis. The first one, based on Feature-wise Linear Modulation (FiLM) layer,

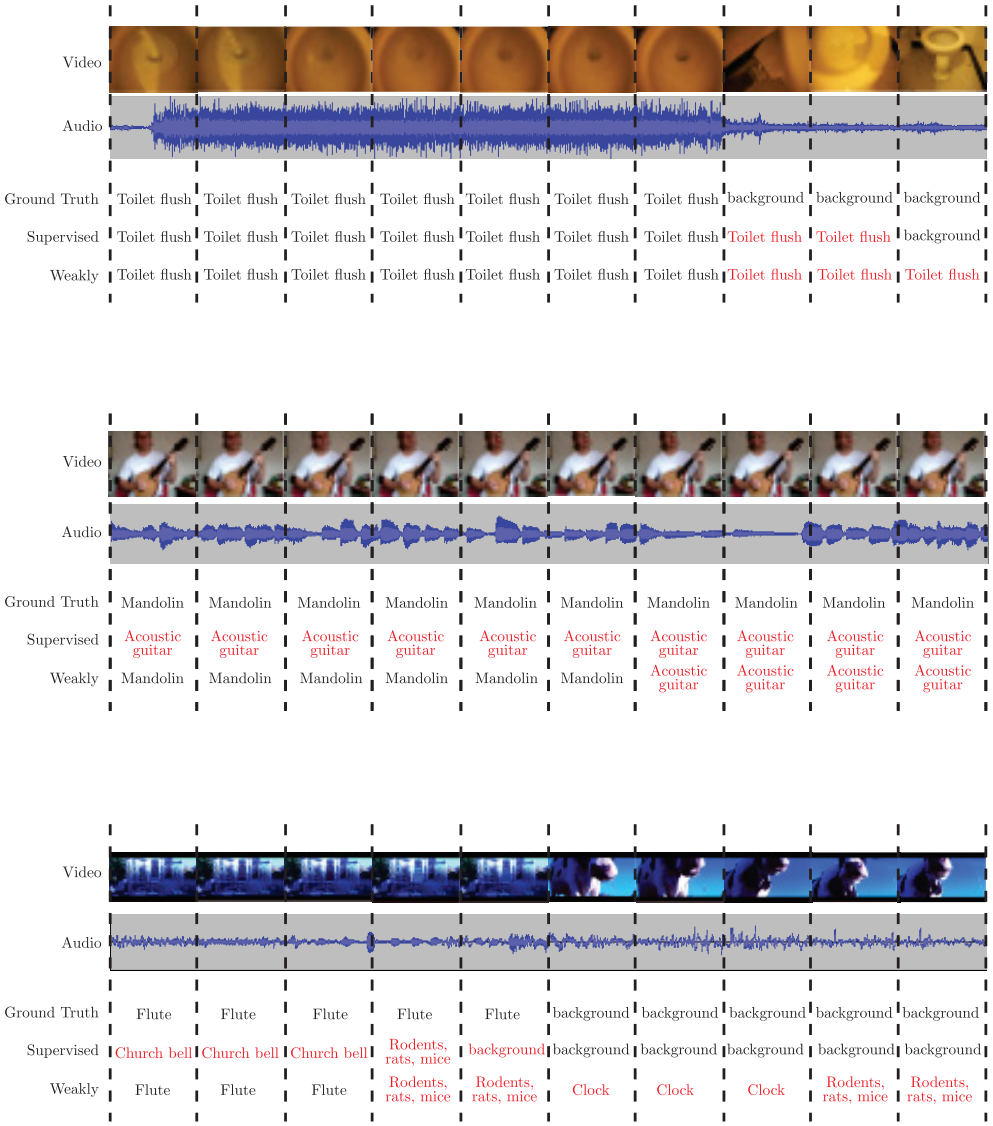


Figure 10.5. Examples of erroneous output estimation for fully-supervised and weakly-supervised tasks.

highlights audio feature maps based on the visual information. The second one, based on Multi-Head Attention (MHA) (inter-modality interactions), highlights time segments of one modality based on the other modality. For a fair comparison, we propose to use the simpler network of Figure 10.6, the conditioning method can be either the FiLM or the multimodal MHA.

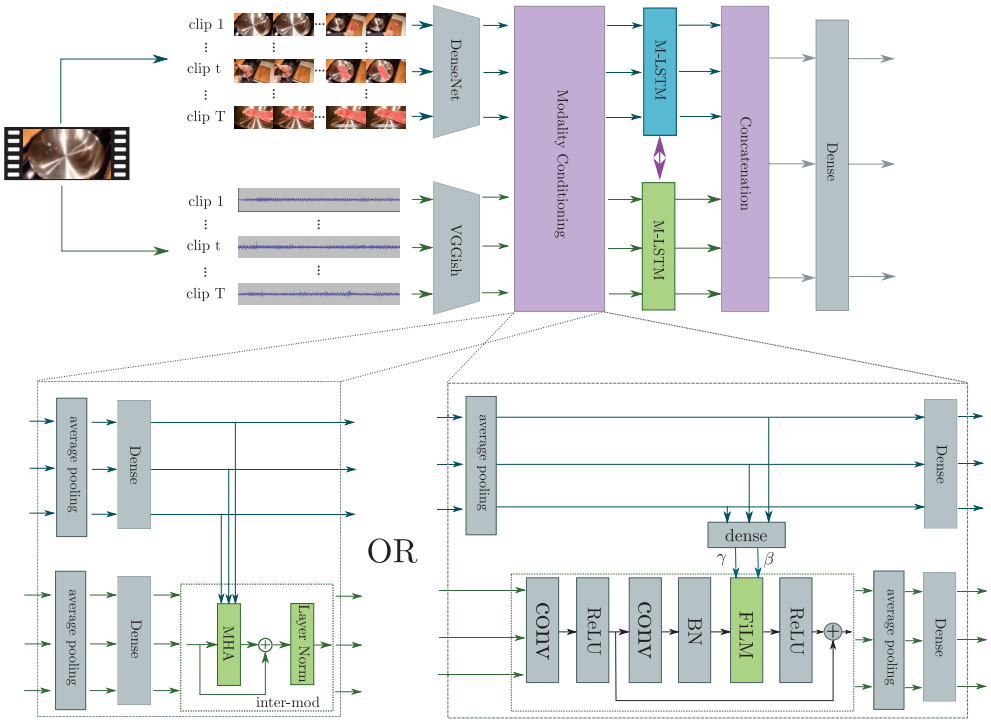


Figure 10.6. Network for comparison of conditioning methods, the conditioning method can be either the FiLM or the multimodal MHA.

In the context of audio-visual event detection, the modality conditioning with multimodal MHA has better results than with FiLM layer (Table 10.5). The advantage of multimodal MHA is that each time segment of one modality interacts with each time segment of the other modality. On the other hand, with FiLM layer, the information of both modalities comes from the same time segment.

	multimodal MHA	FiLM
Accuracy [%]	71.24	68.13

Table 10.5. Comparison of audio conditioning with MHA or FiLM.

10.2.7 Discussion

The temporal aspect is important in detection and classification. A one-second segment does not necessarily comprise enough information to classify an event. Taking into account the context with LSTM and Multi-Head Attention (MHA) is therefore essential. Moreover, an event is present only if it is both visible and audible. The interaction between modalities is necessary to find the segments where the event occurs in the visual and audio domains.

The main error is the temporal detection of the event. Most of the time, the event is correctly classified but the beginning and end of the event are not correct. Better detection will significantly improve results.

The two conditioning methods are performed in different spaces. Feature-wise Linear Modulation (FiLM) influences each feature while MHA influences the time segments. In the detection context, we notice that interactions in temporal space are important. Indeed, FiLM adds to the sound modality the visual information from the same time step but does not take into account the temporal context unlike MHA. Furthermore, the FiLM layer is added to a residual block composed of several convolutional layers, this technique therefore has more parameters and takes longer to train than MHA.

The AVE dataset has stimulated research on audio-visual event detection. However, this database has some drawbacks and its limits have probably been reached. Indeed, some videos include several events but are classified with only one event (although the other events are also part of the classes in the dataset). Moreover, some videos are in both the training and test sets but with a different label. Finally, the resolution of the detection is quite large (1 second). A lot can happen in one second.

10.3 In brief

Summary of Chapter 10

- In this chapter, we introduced a multi-level interactive audio-visual network that efficiently exploit audio and visual information.
- We took into account the intra and inter-modality interaction with the Multi-Head Attention (MHA) mechanism.
- We included the temporal information with a multimodal LSTM. Instead of using two separate traditional LSTMs, one for each modality, we proposed to use multimodal LSTMs where some weights are shared between the visual LSTM and audio LSTM.

Perspective for Chapter 10

- The current models classify events based on audio-visual information, it would be interesting to add the event localization in space as an additional output of the networks.

Chapter 11

Audio-visual event classification and localization

Contents

To conclude this thesis, we evaluate the multimodal networks described in Chapters 9 and 10 with the new audio-visual dataset from Chapter 4. First, we analyze the audio-visual classification performances for the unilabel and multilabel sequences. Then, we present multimodal networks for audio-visual event classification and localization, based on Chapters 6, 9 and 10. An additional output is added in network architectures to address the localization problem, as described in Chapter 6.

As a brief reminder, the new dataset (fully described in Chapter 4) is composed of microphone array recordings (7 microphones) and camera recordings (4 webcams). It includes 2662 unilabel sequences of 3 seconds (one event per

sequence) and 2729 multilabel sequences of 4 seconds (two simultaneous events per sequence). The dataset includes 11 classes.

11.1 Classification on AVECL-UMONS

On one hand, Multi-level Attention Fusion network (MAFnet) from Chapter 9 is used without modification. To simplify the distinction between networks, 'multimodal MAFnet' is the network composed of the modality conditioning with FiLM layer and the fusion with the Modality & temporal attention module. The unimodal network, composed of only dense layers and temporal attention, is named 'unimodal MAFnet'.

On the other hand, the model of Chapter 10, composed of Multi-Head Attentions (MHAs) and multimodal Long Short-Term Memory (LSTM), is used with some modifications. The new database was not designed for event detection. It does not comprise temporal annotation. A class must be estimated for the entire video and not for each segment. As the model is basically designed for event detection, the estimations for each segment are aggregated to get a video-level estimation during the training and testing phases:

$$\widehat{y} = \frac{1}{T}\sum_{t=1}^{T} y_t \tag{11.1}$$

where y_1, ..,y_T are the estimations from the last dense layer of the network for each segment. Indeed,

To simplify the name, 'multimodal MHA' refers to the complete network composed of intra and inter-modality interaction, multimodal-LSTM and concatenation. 'Unimodal MHA' refers to a network composed of only intra-modality interaction and a regular LSTM.

As the unilabel and multilabel sequences of the dataset have different lengths, they are evaluated separately. For each model, the last activation function is a softmax for the unilabel sequences and a sigmoid for the multilabel sequences. The final estimated class is the class with the maximum output estimation for

the unilabel sequences or the classes with an output greater than a threshold of 0.5 for the multilabel sequences.

11.1.1 Feature extraction

Visual feature Visual features are extracted with the same extractor as before (DenseNet [194] pre-trained on ImageNet). The 4 webcam images are concatenated to create a single image. This image is then resized to a 224×224 image and fed to the extractor. The video is divided into $T = 3$ segments for unilabel sequences and into $T = 4$ segments for multilabel sequences. Each segment is one second long. For each segment, we extract the output of the DenseNet last convolutional layer for 16 RGB video frames and then apply global average pooling over the 16 frames.

Audio feature Audio features are extracted with the same audio extractor (VGG-like network [244] pre-trained on AudioSet). As the extractor takes only one microphone channel as input, only the recording of one microphone (micro0) is used. The audio signal is divided into $T = 3$ segments for unilabel sequences and into $T = 4$ segments for multilabel sequences. Each segment is one second long. For each segment, we extract the output of the VGG-like last convolutional layer.

11.1.2 Unilabel performance

Table 11.1 reports the classification performance for the unilabel sequences and compares unimodal and multimodal performances for both models.

As a reminder, Split1 is the random split of data between the training and test sets. Unimodal classification based on sound information is slightly better than the classification based on image, regardless of the network. Moreover, for unilabel classification, the MHA (intra-modality interaction) is better than the temporal attention of MAFnet. The possibility of interaction of each time segment with each other is beneficial for event recognition. Finally, multimodal MAFnet has the best performance while the multimodal MHA fails to

F-score [%]	Split1		Split2	
	MAFnet	MHA	MAFnet	MHA
Image	85.19	86.02	29.16	30.95
Sound	94.35	96.49	85.59	86.78
Both	98.90	95.94	68.35	88.52

Table 11.1. Classification performance for unilabel sequences.

exploit the additional information of each modality.

Split2 tests the generalization capabilities of networks: one subclass from each class is not seen by the network during training. This subclass is then used to test the model. Unimodal classification based on visual information has awful results. The most likely reason is that the 4 webcams have a wide viewpoint of the scene. Therefore, the action is only present on a few pixels. Moreover, since the images from the 4 webcams are concatenated and then resized before extraction, there is probably a great loss of information. The network may base its classification on the room brightness, the location of the person in the room, the person outfit, etc. However, this issue does not occur for the audio modality which has much better results. We notice that the visual modality has a bad influence in the case of the multimodal MAFnet but has no impact on the multimodal MHA.

11.1.3 Multilabel performance

Table 11.2 reports the classification performance for the multilabel sequences and compares unimodal and multimodal performances for both models.

For multilabel classification, similar conclusions can be drawn as for unilabel classification. For Split1, the unimodal MHA has again better results than the temporal attention. Multimodal MAFnet has the best result.

F-score [%]	Split1		Split2	
	MAFnet	MHA	MAFnet	MHA
Image	79.87	95.86	23.24	18.95
Sound	82.10	95.85	55.07	63.78
Both	96.89	94.59	64.87	62.03

Table 11.2. Classification performance for multilabel sequences.

Split2 analyzes the network's ability to classify two classes that never occur together during training. Again, the classification based on visual information has awful performance probably for the same reasons as the unilabel classification. The audio classification is better but not as good as Split1. Models learn to recognize duos of classes instead of each class independently. Finally, the use of both modalities does not improve results for multimodal MHA but has a positive impact for multimodal MAFnet.

11.2 Classification and localization on AVECL-UMONS

The classification models are adapted to the localization task by adding an additional output to the networks.

For multimodal MAFnet, the distinction between the two tasks is implemented before the Modality & temporal attention module. One module is created for each task because the relevant temporal segment and the relevant modality may be different depending on the task (Figure 11.1).

For multimodal MHA, the distinction between the two tasks is done at the level of the last dense layer (Figure 11.2).

In both models, the class is estimated with a classification layer (dense layer with softmax for the unilabel sequences and sigmoid for the multilabel sequences). The localization in the room is estimated with a multi-regression layer (one regression for each class).

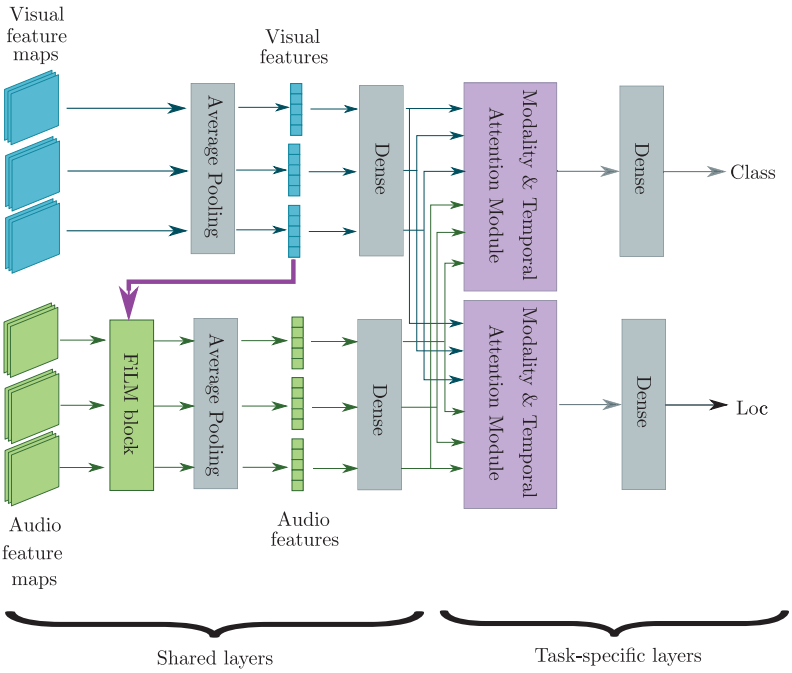


Figure 11.1. Multimodal MAFnet for audio-visual event classification and localization. First, the audio feature maps are modulated by the visual information in the FiLM block. Then, for each subtask, one Modality & temporal attention module highlights the relevant temporal segment and modality. Finally, one dense layer estimates the class as a classification problem and a second dense layer estimates the location in the room as a regression problem.

11.2.1 Feature extraction

The same visual features as the classification are used. Audio features extracted with the VGGish network only include the information of a single microphone. The sound localization can not be performed based on these features. Therefore, new audio features are extracted with the SELDnet model presented in Chapter 6. Features for unilabel and multilabel sequences are extracted by taking the output of the last recurrent layer of the model pretrained on the unilabel and multilabel data, respectively. The entire sequence

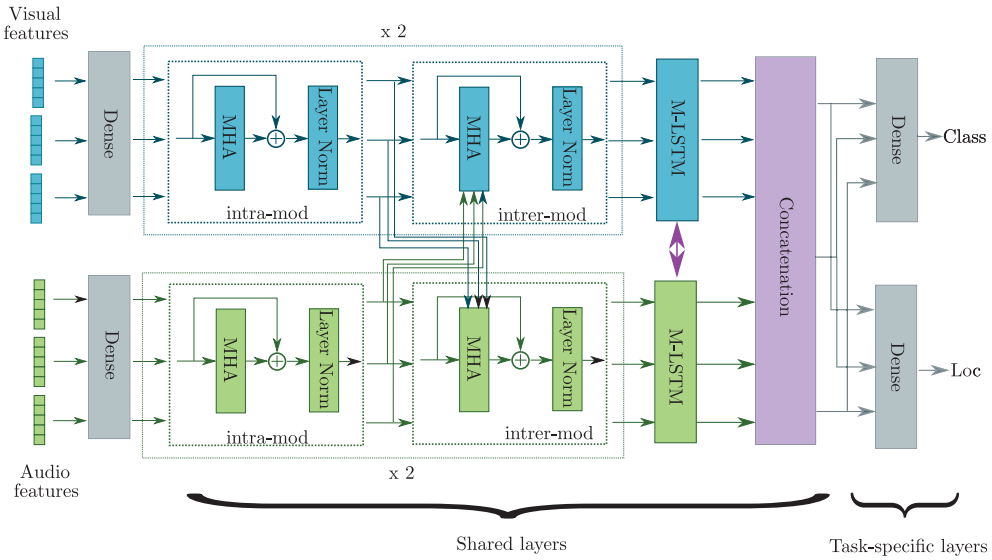


Figure 11.2. Multimodal MHA for audio-visual event classification and localization. Each modality is fed into 2 intra and inter-modality interaction blocks (composed of MHA) and a multimodal LSTM (M-LSTM). The modalities are concatenated. Finally, one dense layer estimates the class as a classification problem and a second dense layer estimates the location in the room as a regression problem.

is processed by the network. As no pooling is applied on the temporal axis, the time scale is unchanged. The output sequence can be divided into T segments that correspond to the T visual segments ($T = 3$ for unilabel sequences and $T = 4$ for multilabel sequences).

11.2.2 Unilabel performance

Table 11.3 reports the classification and localization performance for the unilabel sequences with both multimodal models.

When comparing the different models, the conclusions for the classification performance do not change compared to the previous section, adding the localization task has no impact on the network behavior. However, the perfor-

		Split1		Split2	
		MAFnet	MHA	MAFnet	MHA
Image	F-score	84.85	85.95	31.94	35.59
	DOA error	50.06	21.69	71.87	66.85
Sound	F-score	90.01	89.84	85.54	88.28
	DOA error	37.64	34.24	36.97	34.91
Both	F-score	94.11	90.36	87.23	87.98
	DOA error	24.28	35.97	33.93	36.56

Table 11.3. Classification and localization performance for unilabel sequences.

mances are slightly worse for Split1, especially for models based only on audio information, but no difference or even better results are noticed for Split2.

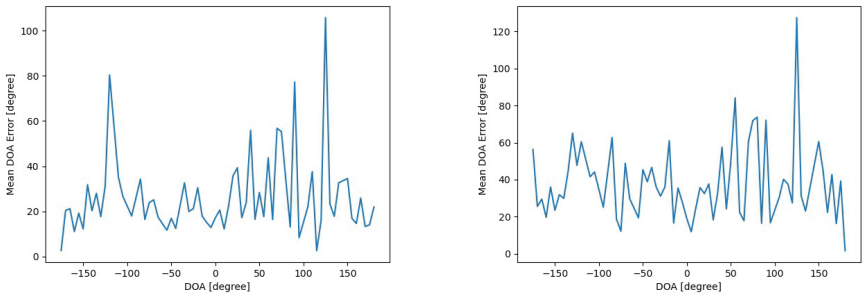
For the localization performances, we observe the same conclusion as for classification. MHA has better results with unimodal information than MAFnet but MAFnet has the best result with multimodal data. Specially, for Split2, the performances based only on visual information are disastrous but the performances based on audio are as good for Split1 as Split2. Again, visual information is useless in the multimodal configuration and therefore, does not improve the results compared to results based on sound information only.

Figure 11.3 shows the DOA error depending on DOA for MAFnet and MHA networks. A large variation can occur between two close DOAs. Therefore, the localization difficulty is not specific to a particular area of the room.

11.2.3 Multilabel performance

Finally, Table 11.3 reports the classification and localization performance for the multilabel sequences with both multimodal models.

In addition to the previous comments, we notice that the multilabel classification performance is slightly worse than the unilabel classification. On the other hand, localization performance is very poor.



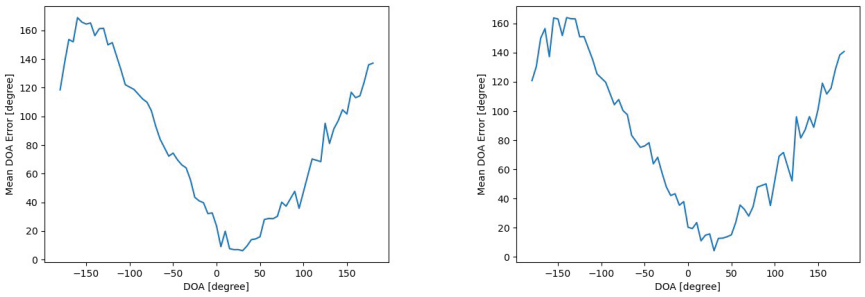
(a) MAFnet **(b)** MHA

Figure 11.3. DOA error depending on DOA for MAFnet and MHA for unilabel sequences.

		Split1		Split2	
		MAFnet	MHA	MAFnet	MHA
Image	F-score	80.78	95.57	21.75	19.20
	DOA error	77.81	77.14	82.83	85.12
Sound	F-score	87.62	89.30	78.32	77.08
	DOA error	77.99	78.08	74.78	74.03
Both	F-score	94.42	88.71	80.44	79.10
	DOA error	76.53	78.1	72.89	74.36

Table 11.4. Classification and localization performance for multilabel sequences.

Figure 11.4 shows the DOA error depending on DOA for MAFnet and MHA networks. It is observed that networks are not able to locate events and, most of the time, estimate a DOA of approximately 30 degrees. The inability to locate multiple events is probably due to the feature extraction phase. Indeed, several averages are performed in the time domain during feature extraction. This is not a problem when only one event occurs. However, when there are two events, the information from the two events is probably mixed and localization can no longer be performed.



(a) MAFnet **(b)** MHA

Figure 11.4. DOA error depending on DOA for MAFnet and MHA for multilabel sequences.

11.3 In brief

Summary of Chapter 11

- In this chapter, we analyzed the classification performances of the models presented in Chapters 9 and 10 on our new dataset. We also adapted these models by adding an output dedicated to the localization problem.
- Multi-Head Attention (MHA) from the intra-modality interaction has a positive impact on event recognition and localization when using only one modality. This network can take into account the temporal context in the sequence which is relevant information whether with image or sound.
- Multi-level Attention Fusion network (MAFnet) is better at exploiting visual and audio information simultaneously. Indeed, the attention mechanism dynamically focuses on modalities rather than always take the same amount of information from the modalities.

Perspective for Chapter 11

- With Split2, results of classification based on visual information are disastrous. The networks classified events based on bad information. There is likely a loss of information due to the position of the cameras and the strong resize of the images. Several strategies would be considered. For example, it would be interesting to treat each camera as a different modality in order to reduce the loss of information during resizing. Another solution would be to crop the interesting part of the video instead of resizing it. However, this solution risks losing the information needed for localization.
- For multilabel sequences, the networks can not distinguish the presence of two events and classify each duo of events as a whole instead of two separate events. It is therefore complicated to locate events separately when they are not able to distinguish them. It would therefore be

interesting to mix the unilabel and multilabel sequences to force the network to recognize each event individually.

- Each sequence is divided into one-second segments. The segment length was constrained by the sound feature extractor (VGGish). Indeed, the extractor network was created to analyze one-second sequences. SELDnet does not apply temporal pooling and the temporal axis is preserved. Any segment size can therefore be used to improve localization performance.

Conclusion

The living beings use all available information to understand a scene or more precisely an event [1]. The Deep Learning (DL) algorithms should use a maximum of information from videos to classify and locate events. During the last years, multimodal data are increasingly used in the context of event recognition. However, there is no single way to jointly exploit audio and visual modalities.

In this chapter, we first summarize the different contributions of this thesis. We then propose several perspectives.

Contributions

AVECL-UMONS dataset There are numerous visual or sound datasets created to address different challenges in classification and detection. However, audio-visual datasets are quite rare. Some datasets, annotated based on visual information, can be exploited as an audio-visual dataset. Unfortunately, some categories are not aurally-manifested or the soundtrack includes irrelevant information such as musical background, oral descriptions of the event, etc. Very recently, several audio-visual datasets were created such as AVE [70] and VGG-Sound [123]. These datasets allow studying the classification of audio-visual events but not the localization. Indeed, most video soundtracks have only one channel and the events occur anywhere without constraint.

We created a new audio-visual dataset in the context of office environments to fill this gap. The dataset is composed of microphone array recordings and four webcam recordings of the same events in two different rooms. It includes unilabel sequences (one event per sequence) and multilabel sequences (two events per sequence) for a total of about 5 hours of recordings. Different metadata

are available such as the event class and its spatial coordinates in the room.

Sound event classification and localization At the time of the new dataset publication, only a single neural network architecture had been proposed in the literature to simultaneously classify and localize events. This network, called SELDnet [101], is based on sound information only. It is composed of several convolutional, recurrent and dense layers.

SELDnet was used to evaluate the audio part of our new dataset. We also slightly changed the architecture to create a more real-time model. We studied different formulations of the localization problem: estimation of x,y coordinates or azimuth angle. The localization should not be class-dependent. It is, therefore, better to use the same regressor for each class rather than training one regressor per class. However, this technique is not applicable to mutlilabel sequences.

The new dataset can be used to classify and localize events. The performance based on sound information only is already good but not perfect. However, the dataset also includes visual data. The visual modality comprises relevant information that can improve results.

Audio-visual fusion and conditioning Although the use of audio-visual data has good results in several tasks, there is no single way to jointly exploit the multimodal data. Inspired by the functioning of the brain [1], the use of audio-visual data must be more than a 'simple' fusion in the neural network but rather several interactions between the audio and the visual paths.

We presented preliminary experiments on audio-visual fusion and audio-visual interaction, named modality conditioning, for event classification. We studied several state-of-the-art fusion techniques (concatenation, addition and Multimodal Compact Bilinear pooling (MCB)) and the modality conditioning with Feature-wise Linear Modulation (FiLM) layers. The FiLM layer highlights some feature maps of one modality based on information of the other modality. As expected, multimodal classification has better performance than unimodal classification. The modality conditioning improves the unimodal classification performance but is not as good as fusion. Modality fusion and conditioning

are complementary techniques. Both have a positive impact on performance and should therefore be used together.

Multi-level Attention Fusion network (MAFnet) The current state-of-the-art fusion techniques exploit all available information of both modalities. However, each modality does not necessarily comprise at all times information relevant for event classification.

To tackle this problem, we proposed Multi-level Attention Fusion network (MAFnet). The network is composed of a Modality & temporal attention module that dynamically highlights relevant time segments and modalities. It also includes the modality conditioning composed of a FiLM layer, introduced in the previous contribution.

Better performances are obtained when the audio and visual data are fused deeper in the network. However, it is important that modalities are not processed independently but rather that they can influence each other. These audio-visual connections create a strong coupling between modalities. The FiLM layer is a conditioning technique but other methods can be implemented to create interaction between visual and audio modalities.

Intra and inter-modality interactions With the release of the AVE dataset, several studies have highlighted the benefit of the interaction between modalities in the detection of audio-visual events during the last year.

Inspired by these works, we proposed a new network architecture that includes several lateral connections between audio and visual paths to couples the processing of the two modalities. First, it models the intra and inter-modality interactions with Multi-Head Attention (MHA). The multimodal MHA creates a soft-alignment between the two modalities to facilitate the detection of both visible and audible events. Indeed, this technique finds the visual and audio segments that comprise related information. Then, the network learns to models temporal contextual information of each segment with multimodal LSTMs. The information can be modality-specific or multimodal.

MHA and multimodal LSTM have an impact on the temporal domain while FiLM is applied in the feature domain. Although feature modulation is a good conditioning strategy, the modality conditioning in the temporal domain is better. FiLM has the disadvantage of taking longer to train.

Audio-visual event classification and localization The two proposed networks only classify events but do not locate them. To conclude this thesis, we adapted the neural networks in order to address the localization problem. We implemented the first proposal for classification and localization of audio-visual events based on Deep Learning (DL) and paves the way for further research.

Perspectives

This thesis brings several improvements for the joint use of multimodal data in the context of audio-visual event recognition and localization. The new proposals are based on methods of the literature. Several approaches have been explored, but some reflections remain to be considered. Here are a few avenues that could be interesting:

- **Temporal labeling of the dataset.** Currently, the dataset does not comprise temporal localization of the events in the sequence. An added bonus for the dataset would be to manually indicate the beginning and end of each event to address the detection task in addition to classification.
- **End-to-end model.** The release of VGGSound dataset [123] provides an audio-visual dataset large enough to train Deep Neural Networks (DNNs) in an end-to-end manner without the prior step of feature extraction. More connections between visual and audio paths within the network could be tested with this dataset, even at early stages.
- **Faster R-CNN** SELDnet have different drawbacks according to the formulation of the localization problem: the association between the detected event and its localization is not done, the impossibility to detect two events with the same label at different locations or two events at

the same place with different classes. Inspired from object detection models, Faster RCNN [149] could be adapted for sound event detection and localization. The model would propose different time regions of interest likely to comprise an event. Then, for each region of interest, the corresponding features would be extracted to classify the event, refine the time region and localize the event in the room. This model would allow a clearer link between the event and its localization and would facilitate the analysis of the results.

- **Training with simulated data** The Deep Neural Network (DNN) training needs a large amount of annotated data. However, the creation of real datasets requires a lot of resources. Different simulation platforms have been created such as HoME [251] or AI2-THOR [252]. Directly annotated simulated data can be extracted from these platforms. These simulated datasets would allow training complex DNNs and then fine-tuning them on smaller real datasets such as our new dataset.
- **Active perception** Currently, the proposed models take all the available visual information as input (the stream of the 4 webcams). However, the human being is able to analyze the scene with the movement of his eyes or even his head which allows him to focus on one object at a time. That's why it would be interesting to implement 'active' localization. The agent would receive the information of one webcam and according to its perceptions, it would be able to slightly modify its point of view. If we take the example of our dataset, an event could take place outside the visual field of webcam 1. The agent would have access to the information from camera 1 and the microphone array. It could estimate that an event takes place outside its visual field based on the sound information. He would then choose another point of view (another webcam) to be able to see the event. More concretely, the active perception could be implemented using the modality attention module. Each webcam would be presented as a different modality instead of presenting the 4 video streams as a single modality. A more "real-time" system can be imagined through reinforcement methods and the use of sliding windows.

Conclusion

Les êtres vivants utilisent toute l'information disponible pour comprendre une scène ou plus précisément un évènement [1]. Les algorithmes d'apprentissage profond devraient utiliser un maximum de l'information présente dans les vidéos pour classifier et localiser les évènements. Durant les dernières années, les données multimodales sont de plus en plus utilisées dans le contexte de la reconnaissance d'évènements. Cependant, il n'y a pas une façon unique de conjointement exploiter les modalités visuelle et sonore.

Dans ce chapitre, nous résumons les différentes contributions de cette thèse. Nous proposons ensuite plusieurs perspectives.

Contributions

La base de données AVECL-UMONS Il y a de nombreuses bases de données visuelles et sonores créées pour résoudre différents problèmes de classification et détection. Cependant, les bases audio-visuelles sont assez rares. Quelques ensembles, annotés sur base de l'information visuelle, peuvent être exploités comme une base audio-visuelle mais certaines catégories ne produisent pas de son particulier ou la bande sonore de la vidéo comprend des informations non pertinentes et trompeuses, par exemple, une musique d'ambiance, une description orale de la vidéo, etc. Très récemment, plusieurs bases de données ont été créées telles que AVE [70] et VGG-Sound [123]. Ces bases permettent d'étudier la classification des évènements audio-visuels mais pas la localisation. En effet, la plupart des vidéos comprennent uniquement un canal audio et les évènements peuvent avoir lieu n'importe où sans restriction.

Nous avons comblé ce manque en créant une nouvelle base de données sur le thème des environnements de bureau. La base de données est composée d'enregistrements d'évènements à l'aide de réseau de microphones et d'enregistrements

des mêmes évènements à l'aide de 4 caméras de type webcam. Les enregistrements ont été réalisés dans deux pièces différentes. La base inclut des séquences uni-étiquettes (un évènement par séquence) et des séquences multi-étiquettes (deux évènements par séquence) pour un total d'environ 5 heures d'enregistrement. Différentes métadonnées sont disponibles telles que la classe de l'évènement and ses coordonnées x,y dans la pièce.

Classification et localisation d'évènements sonores. Au moment de la publication de la base de données, seule une architecture était proposée dans la littérature pour simultanément classifier et localiser des évènements. Ce réseau, nommé SELDnet [101], utilise uniquement l'information sonore. Il est composé de plusieurs couches convolutionnelles, récurrentes et denses.

SELDnet a été utilisé pour évaluer la partie sonore de notre nouvelle base de données. Nous avons également légèrement modifié l'architecture afin de la rendre plus temps-réel. Finalement, nous avons étudié différentes manières de formuler la problématique de localisation: estimer les coordonnées spatiales x,y ou la coordonnée sphérique azimut. Suivant la formulation de la problématique, différents problèmes surviennent. Comme la localisation ne devrait pas être dépendante de la classe de l'évènement, il est meilleur d'utiliser un régresseur unique pour toutes les classes plutôt que d'utiliser un régresseur par classe. Cependant cette technique n'est applicable que pour les séquences uni-étiquettes mais pas pour les séquences multi-étiquettes. Il faut alors ajouter une sortie supplémentaire pour le second évènement et il n'est donc plus possible de savoir quelle localisation correspond à quel évènement.

La nouvelle base de données peut être utilisée pour classifier et localiser des évènements. En se basant uniquement sur l'information sonore, les performances sont déjà bonnes mais pas parfaites. Or, la base de données inclut également des données visuelles contenant de l'information qui permettrait d'améliorer les résultats.

Fusion et conditionnement audio-visuel. Même si l'utilisation de données audio-visuelles donne de bons résultats dans plusieurs tâches, il n'y a pas une façon unique de conjointement exploiter les données multimodales. En

s'inspirant du fonctionnement du cerveau, l'utilisation de données audio-visuelles doit être plus qu'une simple fusion à un certain point du réseau de neurones mais bien plusieurs interactions entre les chemins visuels et sonores.

Nous avons présenté les expériences préliminaires sur la fusion et l'interaction audio-visuelle, appelée conditionnement audio-visuel, dans le contexte de la classification d'évènements. Nous avons analysé plusieurs techniques de fusion de la littérature (concaténation, addition et Multimodal Compact Bilinear pooling (MCB)) ainsi que le conditionnement de modalités avec des couches de modulation linéaire sur les caractéristiques (Feature-wise Linear Modulation (FiLM)). La couche FiLM met en évidence certaines cartes de caractéristique d'une modalité en se basant sur les informations de l'autre modalité. Comme espéré, la classification multimodale a de meilleures performances que la classification unimodale. Le conditionnement de modalités améliore les performances de classification unimodale mais n'est pas aussi bon que la fusion. La fusion et le conditionnement de modalités sont des techniques complémentaires. Elles ont toutes les deux un impact positif sur les performances et devraient donc être utilisées ensemble.

Multi-level Attention Fusion network (MAFnet) Les techniques actuelles de fusion dans la littérature utilisent toute l'information disponible au sein des deux modalités. Cependant, chaque modalité ne comprend pas forcément, à tout moment, de l'information pertinente pour la classification.

Pour aborder ce problème, nous avons proposé un réseau appelé Multi-level Attention Fusion network (MAFnet). Le réseau est composé d'un module d'attention sur les segments de temps et sur les modalités. Ce module donne dynamiquement plus de poids aux segments de temps de chaque modalité qui contiennent de l'information pertinente. Il inclut également la méthode de conditionnement constitué d'une couche FiLM, introduite dans la contribution précédente.

Les performances sont meilleures quand les données audio-visuelles sont fusionnées plus tard, plus profondément dans le réseau. Cependant, il est important de ne pas traiter les données de façon indépendante et de permettre aux modalités de s'influencer mutuellement. La couche FiLM est une technique de conditionnement mais d'autres méthodes peuvent être implémentées pour créer des interactions entre les modalités visuelles et sonores.

Interaction intra et inter-modalités. Avec la publication de la base de données AVE, de plus en plus d'études ont souligné l'intérêt de l'interaction entre les modalités pour la détection d'évènements audio-visuels.

En s'inspirant de ces travaux, nous avons proposé une nouvelle architecture de réseau qui inclut plusieurs connexions entre les chemins visuel et sonore pour coupler le traitement des deux modalités. Premièrement, le réseau modélise les interactions intra et inter-modalités avec un mécanisme d'attention multi-têtes (Multi-Head Attention (MHA))). Le MHA multimodal crée une sorte d'alignement entre les deux modalités pour faciliter la détection d'évènement qui est à la fois visible et audible. En effet, ce mécanisme trouve les segments visuels et sonores qui comprennent de l'information connexe. Ensuite, le modèle apprend à utiliser les informations présentes dans le contexte temporel de chaque segment via des LSTMs multimodaux. L'information peut être spécifique à une modalité mais également multimodale.

L'attention multi-tête et le LSTM multimodal travaillent dans le domaine temporel tandis que le conditionnement FiLM est appliqué au niveau des caractéristiques des modalités. Même si la modulation des caractéristiques est une bonne stratégie, le conditionnement des modalités au niveau temporel est meilleur. De plus, FiLM a le désavantage d'être long à entraîner.

Classification et localisation d'évènements audio-visuels. Les deux réseaux proposés durant la thèse permettent uniquement de classifier les évènements mais pas de les localiser. Pour conclure cette thèse, nous avons donc adapté les réseaux de neurones dans le but d'aborder le problème de localisation en ajoutant une sortie supplémentaire aux modèles. Nous avons donc proposé deux premiers réseaux de référence pour la classification et localisation d'évènement audio-visuel basé sur l'apprentissage profond et ouvert la voie à de nouvelles recherches.

Perspectives

Cette thèse apporte plusieurs améliorations pour l'utilisation conjointe des données multimodales dans le contexte de la reconnaissance et de la localisation d'évènement audio-visuels. Les nouvelles propositions sont basées sur

des techniques de la littérature. Plusieurs approches ont été explorées mais plusieurs pistes peuvent encore être considérées. Voici quelques exemples de pistes qui pourraient être intéressantes:

- **Annotation temporelle de la base de données.** Actuellement, la base de données ne comprend pas de localisation temporelle des évènements. Un point fort supplémentaire pour la base serait de manuellement annoter chaque séquence avec le début et la fin dans le temps de chaque évènement. Ceci permettrait d'aborder la problématique de détection en plus de la classification et localisation.

Modèle de bout en bout. La publication de la base de données VGGSound [123] fournit assez d'exemples pour entraîner des réseaux de neurones profonds de manière bout-à-bout sans passer par une étape indépendante d'extraction de caractéristiques. Cette base de données permettrait donc de tester des architectures avec plus de connexions entre les chemins visuel et sonore, même au niveau des premières couches du réseau.

Faster RCNN. Même si SELDnet a de bons résultats, le réseau à différents inconvénients suivant la formulation du problème de localisation: il n'y a pas d'association claire entre l'évènement détecté et sa localisation dans l'espace, l'impossibilité de détecter deux évènements avec la même classe mais à des emplacements différents ou de détecter deux évènements de différentes classes à la même place. En s'inspirant des modèles de détection d'objets dans les images, Faster RCNN [149] pourrait être adapté pour la détection et la localisation d'évènements sonores. Le modèle pourrait proposer différentes régions temporelles qui comprendraient possiblement un évènement. Ensuite, pour chaque région d'intérêt, les caractéristiques correspondantes pourraient être extraites pour classifier l'évènement, affiner la région temporelle et localiser l'évènement dans la pièce. Ce modèle pourrait permettre un lien plus clair entre chaque évènement et sa localisation dans la pièce. Ceci faciliterait l'analyse des résultats.

Entraînement avec des données simulées. L'entraînement de réseaux de neurones profonds nécessite une grande quantité de données et la création de telles bases de données demande beaucoup de ressources. Actuellement, différentes plates-formes de simulation, par exemple HoME [251] ou AI2-THOR [252], permettent de créer des bases de données simulées. Ces bases simulées permettraient d'entraînement des modèles complexes et ensuite de les adapter sur des bases de données réelles plus petites, comme par exemple notre base.

Perception active. Actuellement, les réseaux proposés prennent toute l'information visuelle disponible comme entrée du réseau (le flux des 4 caméras). Cependant, l'être humain est capable d'analyser une scène grâce aux mouvements des yeux ou encore de la tête afin de se concentrer sur un objet à la fois. L'agent pourrait donc recevoir l'information d'une seule caméra et suivant ce qu'il voit et entend, il pourrait décider de légèrement modifier son point de vue. Si nous prenons l'exemple de notre base de données, un évènement peut avoir lieu en dehors du champ visuel de la webcam 1. L'agent aurait accès aux informations de la caméra 1 et du réseau de microphones. Il pourrait déterminer sur base de l'information sonore qu'un évènement a lieu hors de son champ visuel. Il choisirait alors un autre point de vue (une autre webcam) pour pouvoir voir l'évènement. Plus concrètement, la perception active pourrait être implémentée en utilisant le mécanisme d'attention sur les modalités. Chaque webcam serait présentée comme une modalité différente au lieu de présenter les 4 flux vidéo comme une modalité unique. On pourrait également imaginer utiliser un système plus "temps réel" grâce à des méthodes de renforcement et l'utilisation de fenêtre coulissante.

Appendix A

Publications related to this thesis

A.1 Papers in Conference Proceedings with Peer Review

- Mathilde **BROUSMICHE**, Stéphane DUPONT, Jean Rouat, "Audio-Visual Fusion And Conditioning With Neural Networks For Event Recognition". In: *International Workshop on Machine Learning for Signal Processing*, Pittsburgh, USA, 2019.

- Mathilde **BROUSMICHE**, Jean Rouat, Stéphane DUPONT, "SECL-UMONS Database for Sound Event Classification and Localization". In *IEEE International Conference on Acoustics, Speech, and Signal Processing(ICASSP)* , Barcelona, Spain, 2020.

- Jean-Benoit Delbrouck, Noé Tits, Mathilde **BROUSMICHE**, Stéphane DUPONT, "A Transformer-Based Joint-Encoding for Emotion Recognition and Sentiment Analysis". In: *Proceedings of Second Grand-Challenge and Workshop on Multimodal Language (Challenge-HML)*, Seattle, USA, 2020.

- Mathilde **BROUSMICHE**, Stéphane DUPONT, Jean Rouat, "Intra and Inter-Modality Interactions for Audio-Visual Event Detection". In: *International Workshop on Human-centric Multimedia Analysis (HuMA'20)*, Seattle, USA, 2020.

A.2 Regular Papers in Journals

- Mathilde **BROUSMICHE**, Jean Rouat, Stéphane DUPONT, "Multi-level Attention Fusion Network for Audio-visual Event Recognition". In *Information Fusion)*, Elsevier, 2020. [submitted]

Appendix B

Network details

Glossary

ep	number of ititerations
lr	learning rate
bs	batch size
n_f	number of filters in convolutional layers
k	dimension of kernel in convolutional layers and pooling layers
mom	momentum
n	number of units in recurrent layers
TimeDist	Time distributed layer (apply a layer to every temporal slice of an input)
n_h	number of neurons in dense layers

B.1 SELDnet

Training parameters	
Split	unilabel: train:2178/test:484 / multilabel: train:2179/test:550
Loss	cross-entropy (classification) + 50 × MSE (localization)
Optimizer	Adam (ep:1000, lr:0.001, patience:100, bs:8)
SELDnet	
FFT extractor	sr: 44100Hz, win_size:512, hop_size:256, hamming window
CNN	Conv2D (n_f: 64, k: 3 × 3) BN (mom: 0.99) ReLU Max pooling (k: 1 × 8) Conv2D (n_f:64, k: 3 × 3) BN (mom: 0.99) ReLU Max pooling (k: 1 × 8) Conv2D (n_f:64, k: 3 × 3) BN (mom: 0.99) ReLU Max pooling (k: 1 × 2) Reshape
RNN	Bi-GRU (n:128, tanh) Bi-GRU (n:128, tanh)
output class	TimeDistributed FC (n_h: 128) TimeDistributed FC (n_h: 11) softmax OR sigmoid
output loc	TimeDistributed FC (n_h: 128) TimeDistributed FC (n_h: 2 × 11) linear

B.2 Fusion and conditioning

Image/Sound feature extractor	DenseNet201 [194]/Mel-band + CNN [240]
Split	cross-test 6 folds (train:80/val:20/test:20)
Loss	cross-entropy
Optimizer	except MCB: SGD (ep:300, lr:0.001, patience:50, bs:10) for MCB:SGD (ep:500, lr:0.005, patience:50, bs:10)

Fusion at 1st level		Fusion at 2nd level		Fusion at 3rd level	
multi	concat FC (n_h:512) BN (mom:0.999) ReLU FC (n_h:10) BN (mom:0.999) softmax	image	FC (n_h:512) BN (mom:0.999) ReLU	image	FC (n_h:512) BN (mom:0.999) ReLU FC (n_h:512) BN (mom:0.999) softmax
		sound	FC (n_h:512) BN (mom:0.999) ReLU		
		mult	concat OR + OR MCB FC (n_h:10) BN (mom:0.999) softmax	sound	FC (n_h:512) BN (mom:0.999) ReLU FC (n_h:10) BN (mom:0.999) softmax
				multi	+

Conditioning network	
FiLM generator (γ, β)	FC (n_h:2 × 512)
Residual Block	Conv2D (n_f:512, k: 1 × 1) ReLU Conv2D (n_f:512, k: 3 × 3) BN (mom:0.999) γ x + β ReLU residual connection
output	FC (n_h: 10) - BN (mom:0.999) - softmax

B.3 Multi-level Attention Fusion network (MAFnet)

Image/Sound feature extractor		DenseNet201 [194]/VGGish [240]
Split		train:3268/val:390/test:394
Loss		cross-entropy
Optimizer		Adam (ep:300, lr:0.001, patience:50, bs:32)
MAFnet		
image	FiLM generator (γ, β)	FC (n_h: 2 $\times$ 512)
		TimeDistributed FC (n_h: 10) BN (mom:0.999) ReLU
sound	Residual Block	Conv2D (n_f:512, k: 1 $\times$ 1) ReLU Conv2D (n_f:512, k: 3 $\times$ 3) BN (mom:0.999) γ x + β ReLU residual connection
		TimeDistributed FC (n_h: 10) BN (mom:0.999) ReLU
multi	Modality and temporal attention	FC (n_h: 512) ReLU FC (n_h: 1) softmax score $\times$ segment concat AND reduce sum
	output	FC (n_h: 28) - BN (mom:0.999) - softmax

B.4 Intra and inter modality interactions

Image/Sound feature extractor	DenseNet201 [194]/VGGish [240]
Split	train:3339/val:402/test:402
Loss	cross-entropy
Optimizer	Adam (ep:50, lr:0.001, bs:32)

Intra and inter-modality interaction + multimodal LSTM		
	image	sound
	TimeDist FC (n_h: 2 × 512) ReLU	TimeDist FC (n_h: 2 × 512) ReLU
intra-modality	MHA (n_h:512, head:2) residual connection Layer Norm	MHA (n_h:512, head:2) residual connection Layer Norm
inter-modality	MHA (n_h:512, head:8) residual connection Layer Norm	
intra-modality	MHA (n_h:512, head:2) residual connection Layer Norm	MHA (n_h:512, head:2) residual connection Layer Norm
inter-modality	MHA (n_h:512, head:8) residual connection Layer Norm	
multimodal LSTM	Bi-LSTM (n: 512)	
Fusion	concat	
output	TimeDist FC (n_h: 28+1) softmax	

Bibliography

[1] E. B. Goldstein and J. Brockmole, *Sensation and perception*. Cengage Learning, 2016.

[2] H. McGurk and J. MacDonald, "Hearing lips and seeing voices", *Nature*, vol. 264, no. 5588, pp. 746–748, 1976.

[3] L. Shams and A. R. Seitz, "Benefits of multisensory learning", *Trends in Cognitive Sciences*, vol. 12, no. 11, pp. 411–417, 2008.

[4] M. H. Giard and F. Peronnet, "Auditory-visual integration during multimodal object recognition in humans: a behavioral and electrophysiological study", *Journal of Cognitive Neuroscience*, vol. 11, no. 5, pp. 473–490, 1999.

[5] J. Driver and T. Noesselt, "Multisensory interplay reveals crossmodal influences on 'sensory-specific' brain regions, neural responses, and judgments", *Neuron*, vol. 57, no. 1, pp. 11–23, 2008.

[6] L. Shams and R. Kim, "Crossmodal influences on visual perception", *Physics of Life Reviews*, vol. 7, no. 3, pp. 269–284, 2010.

[7] M. Ursino, C. Cuppini, and E. Magosso, "Neurocomputational approaches to modelling multisensory integration in the brain: a review", *Neural Networks*, vol. 60, pp. 141–165, 2014.

[8] T. Lu and T.-H. Chao, *Advances in Pattern Recognition Research*. Nova Science Publishers, Incorporated, 2018.

[9] S. Laraba, "Deep Learning for Skeleton-Based Human Action Recognition", Ph.D. dissertation, University of Mons, 2020.

[10] O. Seddati, "Reconnaissance et Recherche de Données Multimédia par les Réseaux de Neurones Profonds", Ph.D. dissertation, University of Mons, 2018.

[11] E. Ennadifi, S. Laraba, D. Vincke, B. Mercatoris, and B. Gosselin, "Wheat diseases classification and localization using convolutional neural networks and gradcam visualization", in *International Conference on Intelligent Systems and Computer Vision (ISCV)*. IEEE, 2020, pp. 1–5.

[12] P. Kong, M. Mancas, N. Thuon, S. Kheang, and B. Gosselin, "Do deep-learning saliency models really model saliency?" in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 2331–2335.

[13] A. Moreau, M. Mancas, and T. Dutoit, "Unsupervised depth prediction from monocular sequences: Improving performances through instance segmentation", in *Conference on Computer and Robot Vision (CRV)*. IEEE, 2020, pp. 54–61.

[14] D. Yu and L. Deng, *AUTOMATIC SPEECH RECOGNITION*. Springer, 2016.

[15] G. Pironkov, "Acoustic Modelling using Deep Neural Networks for Automatic Speech Recognition", Ph.D. dissertation, University of Mons, 2017.

[16] U. Kamath, J. Liu, and J. Whitaker, *Deep learning for NLP and speech recognition*. Springer, 2019, vol. 84.

[17] P. Koehn, *Neural machine translation*. Cambridge University Press, 2020.

[18] J.-B. Delbrouck, "Grounding and pragmatics for multimodal Human-

Machine interaction", Ph.D. dissertation, University of Mons, 2020.

[19] S. K. Zhou, H. Greenspan, and D. Shen, *Deep learning for medical image analysis*. Academic Press, 2017.

[20] V. Delvigne, T. Dutoit, L. Ris, H. Wannous, and J. Vandenborre, "An innovative neurofeedback for children with adhd using virtual reality [abstract]", in *4th HBP Student Conference on Interdisciplinary Brain Research*, 2020.

[21] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[22] Y. Ning, S. He, Z. Wu, C. Xing, and L.-J. Zhang, "A review of deep learning based speech synthesis", *Applied Sciences*, vol. 9, no. 19, p. 4050, 2019.

[23] N. Tits, K. El Haddad, and T. Dutoit, "Neural speech synthesis with style intensity interpolation: A perceptual analysis", in *ACM/IEEE International Conference on Human-Robot Interaction*, 2020, pp. 485–487.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[25] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks", in *Proceedings of the fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[26] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized MLP architectures of neural networks", *International Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, pp. 111–122, 2011.

[27] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning", in *International Work-*

shop on Artificial Neural Networks. Springer, 1995, pp. 195–201.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors", *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[29] J. S. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters", in *Advances in Neural Information Processing Systems*, 1990, pp. 211–217.

[30] C. Lemaréchal, "Cauchy and the gradient method", *Doc Math Extra*, vol. 251, p. 254, 2012.

[31] H. B. Curry, "The method of steepest descent for non-linear minimization problems", *Quarterly of Applied Mathematics*, vol. 2, no. 3, pp. 258–261, 1944.

[32] O. H. Rodríguez and J. M. Lopez Fernandez, "A semiotic reflection on the didactics of the chain rule", *The Mathematics Enthusiast*, vol. 7, no. 2, pp. 321–332, 2010.

[33] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning", in *Advances in Neural Information Processing Systems*, 2008, pp. 161–168.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

[35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[36] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization", *arXiv preprint arXiv:1506.06579*, 2015.

[37] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient

flow in recurrent nets: the difficulty of learning long-term dependencies", 2001.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation", *arXiv preprint arXiv:1406.1078*, 2014.

[40] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks", *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[41] Z. Deng, "Survey on various approaches of saliency detection", in *International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*. IEEE, 2019, pp. 358–363.

[42] S. Jetley, N. A. Lord, N. Lee, and P. H. Torr, "Learn to pay attention", in *International Conference on Learning Representations*, 2018.

[43] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3156–3164.

[44] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate", *arXiv preprint arXiv:1409.0473*, 2014.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin, "Attention is all you need", in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5998–6008.

[46] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel,

and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention", in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[47] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4651–4659.

[48] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention", in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2204–2212.

[49] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation", in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015, pp. 1412–1421.

[50] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading", in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2016, pp. 551–561.

[51] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", in *International Conference on Machine Learning*, 2015, pp. 448–456.

[52] S. Poria, E. Cambria, N. Howard, G.-B. Huang, and A. Hussain, "Fusing audio, visual and textual clues for sentiment analysis from multimodal content", *Neurocomputing*, vol. 174, pp. 50–59, 2016.

[53] C.-H. Wu, J.-C. Lin, and W.-L. Wei, "Survey on audiovisual emotion recognition: databases, features, and data fusion strategies", *APSIPA Transactions on Signal and Information Processing*, vol. 3, 2014.

[54] G. Potamianos, C. Neti, J. Luettin, and I. Matthews, "Audio-visual automatic speech recognition: An overview", *Issues in Visual and Audio-*

visual Speech Processing, vol. 22, p. 23, 2004.

[55] A. Gabbay, A. Ephrat, T. Halperin, and S. Peleg, "Seeing through noise: Visually driven speaker separation and enhancement", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 3051–3055.

[56] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation", *ACM Transactions on Graphics (TOG)*, pp. 1–11, 2018.

[57] J. R. Hershey and J. R. Movellan, "Audio vision: Using audio-visual synchrony to locate sounds", in *Advances in Neural Information Processing Systems*, 2000, pp. 813–819.

[58] A. Senocak, T.-H. Oh, J. Kim, M.-H. Yang, and I. So Kweon, "Learning to localize sound source in visual scenes", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4358–4366.

[59] K. Sriskandaraja, V. Sethu, and E. Ambikairajah, "Deep siamese architecture based replay detection for secure voice biometric." in *Interspeech*, 2018, pp. 671–675.

[60] L. Chen, S. Srivastava, Z. Duan, and C. Xu, "Deep cross-modal audio-visual generation", in *Proceedings of the on Thematic Workshops of ACM Multimedia*, 2017, pp. 349–357.

[61] D. Ramachandram and G. W. Taylor, "Deep multimodal learning: A survey on recent advances and trends", *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 96–108, 2017.

[62] W. Guo, J. Wang, and S. Wang, "Deep multimodal representation learning: A survey", *IEEE Access*, vol. 7, pp. 63 373–63 394, 2019.

[63] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency, "Tensor Fusion Network for Multimodal Sentiment Analysis", in *Proceedings of the Conference on Empirical Methods in Natural Language*. Association for Computational Linguistics, 2017, pp. 1103–1114.

[64] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, "Compact bilinear pooling", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 317–326.

[65] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding", in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016, p. 457–468.

[66] J.-B. Delbrouck and S. Dupont, "Multimodal compact bilinear pooling for multimodal neural machine translation", *arXiv preprint arXiv:1703.08084*, 2017.

[67] Z. Yu, J. Yu, J. Fan, and D. Tao, "Multi-modal factorized bilinear pooling with co-attention learning for visual question answering", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1821–1830.

[68] J. Ramaswamy, "What makes the sound?: A dual-modality interacting network for audio-visual event localization", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 4372–4376.

[69] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[70] Y. Tian, J. Shi, B. Li, Z. Duan, and C. Xu, "Audio-visual event localization in unconstrained videos", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 247–263.

[71] H. Zhu, M. Luo, R. Wang, A. Zheng, and R. He, "Deep audio-visual learning: A survey", *arXiv preprint arXiv:2001.04758*, 2020.

[72] E. Petajan, "Automatic lipreading to enhance speech recognition", in *Proceedings of the Global Telecommunications Conference*. IEEE, 1984, pp. 265–272.

[73] B. P. Yuhas, M. H. Goldstein, and T. J. Sejnowski, "Integration of acoustic and visual speech signals using neural networks", *IEEE Communications Magazine*, vol. 27, no. 11, pp. 65–71, 1989.

[74] G. Potamianos, "Audio-visual automatic speech recognition and related bimodal speech technologies: A review of the state-of-the-art and open problems", in *IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2009, pp. 22–22.

[75] A. P. Kandagal and V. Udayashankara, "Automatic bimodal audiovisual speech recognition: A review", in *International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE, 2014, pp. 940–945.

[76] I. Addarrazi, H. Satori, and K. Satori, "A follow-up survey of audiovisual speech integration strategies", in *Embedded Systems and Artificial Intelligence*. Springer, 2020, pp. 635–643.

[77] S. Dupont and J. Luettin, "Audio-visual speech modeling for continuous speech recognition", *IEEE Transactions on Multimedia*, vol. 2, no. 3, pp. 141–151, 2000.

[78] A. Adjoudani and C. Benoit, "On the integration of auditory and visual parameters in an hmm-based asr", in *Speechreading by humans and machines*. Springer, 1996, pp. 461–471.

[79] S. Haq and P. J. Jackson, "Multimodal emotion recognition", in *Machine audition: principles, algorithms and systems*. IGI Global, 2011, pp. 398–423.

[80] A. Nagrani, S. Albanie, and A. Zisserman, "Seeing voices and hearing faces: Cross-modal biometric matching", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8427–8436.

[81] R. Wang, H. Huang, X. Zhang, J. Ma, and A. Zheng, "A novel distance learning for elastic cross-modal audio-visual matching", in *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2019, pp. 300–305.

[82] R. Arandjelovic and A. Zisserman, "Look, listen and learn", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 609–617.

[83] ——, "Objects that sound", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 435–451.

[84] A. Owens and A. A. Efros, "Audio-visual scene analysis with self-supervised multisensory features", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 631–648.

[85] E. Kidron, Y. Y. Schechner, and M. Elad, "Pixels that sound", in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2005, pp. 88–95.

[86] K. Li, J. Ye, and K. A. Hua, "What's making that sound?" in *Proceedings of the 22nd ACM International Conference on Multimedia*, 2014, pp. 147–156.

[87] B. Korbar, D. Tran, and L. Torresani, "Co-training of audio and video representations from self-supervised temporal synchronization", *arXiv preprintarXiv:1807.00230*, 2018.

[88] S. Parekh, S. Essid, A. Ozerov, N. Q. Duong, P. Pérez, and G. Richard, "Weakly supervised representation learning for unsynchronized audio-visual events." in *Computer Vision and Pattern Recognition Workshops*,

2018, pp. 2518–2519.

[89] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video", in *Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 892–900.

[90] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba, "Ambient sound provides supervision for visual learning", in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 801–816.

[91] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[92] K. J. Piczak, "ESC: Dataset for environmental sound classification", in *Proceedings of the 23rd ACM International Conference on Multimedia*, 2015, pp. 1015–1018.

[93] H. Dubey, D. Emmanouilidou, and I. J. Tashev, "CURE dataset: Ladder networks for audio event classification", in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*. IEEE, 2019, pp. 1–6.

[94] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research", in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 1041–1044.

[95] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection", in *EUSIPCO*. IEEE, 2016, pp. 1128–1132.

[96] S. Gharib, K. Drossos, E. Fagerlund, and T. Virtanen, "VOICe: A sound event detection dataset for generalizable domain adaptation", *arXiv preprint arXiv:1911.07098*, 2019.

[97] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo, "Acoustic event detection and classification in smart-room environments: Evaluation of CHIL project systems", *Cough*, vol. 65, no. 48, p. 5, 2006.

[98] H. Meraoubi and B. Boudraa, "Multimicrophone source localization data base (MUSLOD)", in *Second World Conference on Complex Systems*. IEEE, 2014, pp. 604–608.

[99] G. Lathoud, J.-M. Odobez, and D. Gatica-Perez, "AV16.3: an audio-visual corpus for speaker localization and tracking", in *International Workshop on Machine Learning for Multimodal Interaction*. Springer, 2004, pp. 182–195.

[100] J. K. Nielsen, J. R. Jensen, S. H. Jensen, and M. G. Christensen, "The single and multichannel audio recordings database (SMARD)", in *International Workshop on Acoustic Signal Enhancement*. IEEE, 2014, pp. 40–44.

[101] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks", *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, 2018.

[102] S. Adavanne, A. Politis, and T. Virtanen, "A multi-room reverberant dataset for sound event localization and detection", *arXiv preprint arXiv:1905.08546*, 2019.

[103] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[104] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective", *International Journal of Computer Vision*, vol. 111, no. 1,

pp. 98–136, 2015.

[105] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context", in *European Conference on Computer Vision*. Springer, 2014, pp. 740–755.

[106] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "YouTube-8M: A large-scale video classification benchmark", *arXiv preprint arXiv:1609.08675*, 2016.

[107] J. Carreira, E. Noland, C. Hillier, and A. Zisserman, "A short note on the kinetics-700 human action dataset", *arXiv preprint arXiv:1907.06987*, 2019.

[108] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition", in *International Conference on Computer Vision*. IEEE, 2011, pp. 2556–2563.

[109] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild", *arXiv preprint arXiv:1212.0402*, 2012.

[110] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1725–1732.

[111] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach", in *Proceedings of the 17th International Conference on Pattern Recognition(ICPR)*, vol. 3. IEEE, 2004, pp. 32–36.

[112] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, "The "Something Something" video database for learning and evaluating visual common sense." in *International Conference on Computer*

Vision (ICCV), vol. 1, no. 4, 2017, p. 5.

[113] D. Damen, H. Doughty, G. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price *et al.*, "The epic-kitchens dataset: Collection, challenges and baselines", *IEEE Computer Architecture Letters*, no. 01, pp. 1–1, 2020.

[114] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding", in *European Conference on Computer Vision*. Springer, 2016, pp. 510–526.

[115] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah, "The THUMOS challenge on action recognition for videos "in the wild"", *Computer Vision and Image Understanding*, vol. 155, pp. 1–23, 2017.

[116] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970.

[117] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, "Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5296–5305.

[118] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar *et al.*, "Ava: A video dataset of spatio-temporally localized atomic visual actions", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6047–6056.

[119] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang,

"Youtube-VOS: A large-scale video object segmentation benchmark", *arXiv preprint arXiv:1809.03327*, 2018.

[120] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 724–732.

[121] M. Monfort, A. Andonian, B. Zhou, K. Ramakrishnan, S. A. Bargal, T. Yan, L. Brown, Q. Fan, D. Gutfreund, C. Vondrick *et al.*, "Moments in time dataset: one million videos for event understanding", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 502–508, 2019.

[122] M. Monfort, K. Ramakrishnan, A. Andonian, B. A. McNamara, A. Lascelles, B. Pan, D. Gutfreund, R. Feris, and A. Oliva, "Multi-moments in time: Learning and interpreting models for multi-action video understanding", *arXiv preprint arXiv:1911.00232*, 2019.

[123] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman, "VGGSound: A large-scale audio-visual dataset", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 721–725.

[124] A. Pieropan, G. Salvi, K. Pauwels, and H. Kjellström, "Audio-visual classification and detection of human manipulation actions", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3045–3052.

[125] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks", in *IJCNN*. IEEE, 2015, pp. 1–7.

[126] ——, "Multi-label vs. combined single-label sound event detection with deep neural networks", in *23rd European Signal Processing Conference (EUSIPCO)*. IEEE, 2015, pp. 2551–2555.

[127] I.-Y. Jeong, S. Lee, Y. Han, and K. Lee, "Audio event detection using multiple-input convolutional neural network", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017.

[128] H. Liu, X. Wang, F.-Q. Guan, and J.-S. Hu, "Convolutional neural networks with multi-task loss for polyphonic sound event detection", in *Proceedings of the International Conference on Computer Science and Application Engineering*. ACM, 2018, p. 87.

[129] P. Zinemanas, P. Cancela, and M. Rocamora, "End-to-end convolutional neural networks for sound event detection in urban environments", in *Proceedings of the Conference of Open Innovations Association FRUCT*. FRUCT Oy, 2019, p. 74.

[130] M. Zöhrer and F. Pernkopf, "Gated recurrent networks applied to acoustic scene classification and acoustic event detection", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016.

[131] S. Adavanne, G. Parascandolo, P. Pertilä, T. Heittola, and T. Virtanen, "Sound event detection in multichannel audio using spatial and harmonic features", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016.

[132] J. Zhou, "Sound event detection in multichannel audio LSTM network", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017.

[133] H.-G. Kim and J. Y. Kim, "Acoustic event detection in multichannel audio using gated recurrent neural networks with high-resolution spectral features", *ETRI Journal*, vol. 39, no. 6, pp. 832–840, 2017.

[134] R. Lu, Z. Duan, and C. Zhang, "Multi-scale recurrent neural network for sound event detection", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 131–135.

[135] M. Valenti, D. Tonelli, F. Vesperini, E. Principi, and S. Squartini, "A

neural network approach for sound event detection in real life audio", in *25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 2754–2758.

[136] T. H. Vu and J.-C. Wang, "Acoustic scene and event recognition using recurrent neural networks", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, vol. 2016, 2016.

[137] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings", in *ICASSP*. IEEE, 2016, pp. 6440–6444.

[138] R. Lu and Z. Duan, "Bidirectional GRU for sound event detection", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017.

[139] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[140] S. Adavanne, P. Pertilä, and T. Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network", in *ICASSP*. IEEE, 2017, pp. 771–775.

[141] J. Ma, R. Wang, W. Ji, H. Zheng, E. Zhu, and J. Yin, "Relational recurrent neural networks for polyphonic sound event detection", *Multimedia Tools and Applications*, pp. 1–19, 2019.

[142] S. Adavanne and T. Virtanen, "A report on sound event detection with different binaural features", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017.

[143] S. Adavanne, A. Politis, and T. Virtanen, "Multichannel sound event detection using 3D convolutional neural networks for learning interchannel features", in *International Joint Conference on Neural Networks*

(IJCNN). IEEE, 2018, pp. 1–7.

[144] T. Iqbal, Y. Xu, Q. Kong, and W. Wang, "Capsule routing for sound event detection", in *26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 2255–2259.

[145] Y. Liu, J. Tang, Y. Song, and L. Dai, "A capsule based approach for polyphonic sound event detection", in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018, pp. 1853–1857.

[146] F. Vesperini, L. Gabrielli, E. Principi, and S. Squartini, "Polyphonic sound event detection by using capsule neural networks", *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 310–322, 2019.

[147] K.-W. Liang, Y.-H. Tseng, and P.-C. Chang, "Parallel capsule neural networks for sound event detection", in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 1933–1936.

[148] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules", in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 3856–3866.

[149] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks", in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[150] K. Wang, L. Yang, and B. Yang, "Audio Event Detection and classification using extended R-FCN Approach", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017.

[151] C.-C. Kao, W. Wang, M. Sun, and C. Wang, "R-CRNN: Region-based convolutional recurrent neural network for audio event detection", in *Interspeech*, 2018.

[152] P. Pham, J. Li, J. Szurley, and S. Das, "Eventness: Object detection on spectrograms for temporal localization of audio events", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2491–2495.

[153] Y. Kiyokawa, S. Mishima, T. Toizumi, K. Sagi, R. Kondo, and T. Nomura, "Sound event detection with ResNet and self-mask module for DCASE 2019 task 4", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2019.

[154] A. Nasiri, Y. Cui, Z. Liu, J. Jin, Y. Zhao, and J. Hu, "AudioMask: Robust Sound Event Detection Using Mask R-CNN and Frame-Level Classifier", in *IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2019, pp. 485–492.

[155] Y.-H. Shen, K.-X. He, and W.-Q. Zhang, "Learning how to listen: A temporal-frequential attention model for sound event detection", *arXiv preprint arXiv:1810.11939*, 2018.

[156] J. Zhang, W. Ding, J. Kang, and L. He, "Multi-scale time-frequency attention for acoustic event detection", *arXiv preprint arXiv:1904.00063*, 2019.

[157] S. Chakrabarty and E. A. Habets, "Broadband DOA estimation using convolutional neural networks trained with noise signals", in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 136–140.

[158] S. Adavanne, A. Politis, and T. Virtanen, "Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network", in *26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1462–1466.

[159] L. Perotin, A. Défossez, E. Vincent, R. Serizel, and A. Guérin, "Regression versus classification for neural network based audio source localiza-

tion", in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 343–347.

[160] Z. Tang, J. D. Kanu, K. Hogan, and D. Manocha, "Regression and classification for direction-of-arrival estimation with convolutional recurrent neural networks", *arXiv preprint arXiv:1904.08452*, 2019.

[161] S. Chakrabarty and E. A. Habets, "Multi-speaker DOA estimation using deep convolutional networks trained with noise signals", *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 8–21, 2019.

[162] X. Xiao, S. Zhao, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, "A learning-based approach to direction of arrival estimation in noisy and reverberant environments", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 2814–2818.

[163] F. Vesperini, P. Vecchiotti, E. Principi, S. Squartini, and F. Piazza, "A neural network based algorithm for speaker localization in a multi-room environment", in *IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2016, pp. 1–6.

[164] L. Perotin, R. Serizel, E. Vincent, and A. Guérin, "CRNN-based multiple DoA estimation using acoustic intensity features for Ambisonics recordings", *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 22–33, 2019.

[165] ——, "CRNN-based joint azimuth and elevation localization with the Ambisonics intensity vector", in *16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2018, pp. 241–245.

[166] J. Vera-Diaz, D. Pizarro, and J. Macias-Guarasa, "Towards end-to-end acoustic localization using deep learning: From audio signals to source position coordinates", *Sensors*, vol. 18, no. 10, p. 3418, 2018.

[167] H. Sundar, W. Wang, M. Sun, and C. Wang, "Raw waveform based

end-to-end deep convolutional network for spatial localization of multiple acoustic sources", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 4642–4646.

[168] R. Takeda and K. Komatani, "Sound source localization based on deep neural networks with directional activate function exploiting phase information", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 405–409.

[169] H. Tsuzuki, M. Kugler, S. Kuroyanagi, and A. Iwata, "An approach for sound source localization by complex-valued neural network", *IEICE Transactions on Information and Systems*, vol. 96, no. 10, pp. 2257–2265, 2013.

[170] O. Bialer, N. Garnett, and T. Tirer, "Performance advantages of deep neural networks for angle of arrival estimation", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3907–3911.

[171] P. Vecchiotti, N. Ma, S. Squartini, and G. J. Brown, "End-to-end binaural sound localisation from the raw waveform", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 451–455.

[172] P. Pertilä and M. Parviainen, "Time difference of arrival estimation of speech signals using deep neural networks with integrated time-frequency masking", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 436–440.

[173] R. Takeda and K. Komatani, "Unsupervised adaptation of deep neural networks for sound source localization using entropy minimization", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2217–2221.

[174] C. Pang, H. Liu, and X. Li, "Multitask learning of time-frequency CNN

for sound source localization", *IEEE Access*, vol. 7, pp. 40 725–40 737, 2019.

[175] T. T. N. Nguyen, W.-S. Gan, R. Ranjan, and D. L. Jones, "Robust source counting and doa estimation using spatial pseudo-spectrum and convolutional neural network", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[176] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, "Overview and evaluation of sound event localization and detection in DCASE 2019", *arXiv preprint arXiv:2009.02792*, 2020.

[177] S. Kapka and M. Lewandowski, "Sound source detection, localization and classification using consecutive ensemble of CRNN models", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2019.

[178] Y. Cao, Q. Kong, T. Iqbal, F. An, W. Wang, and M. D. Plumbley, "Polyphonic sound event detection and localization using a two-stage strategy", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2019.

[179] W. Xue, T. Ying, Z. Chao, and D. Guohong, "Multi-beam and multi-task learning for joint sound event detection and localization", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2019.

[180] T. Hirvonen, "Classification of spatial audio location and content using convolutional neural networks", in *Audio Engineering Society Convention 138*. Audio Engineering Society, 2015.

[181] S. P. Chytas and G. Potamianos, "Hierarchical detection of sound events and their localization using convolutional neural networks with adaptive thresholds", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2019.

[182] S. Park, "TrellisNet-based architecture for sound event localization and detection with reassembly learning", *Detection and Classification of*

Acoustic Scenes and Events (DCASE), 2019.

[183] S. Leung and Y. Ren, "Spectrum combination and convolutional recurrent neural networks for joint localization and detection of sound events", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2019.

[184] T. N. T. Nguyen, D. L. Jones, R. Ranjan, S. Jayabalan, and W. S. Gan, "A two-step system for sound event localization and detection", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2019.

[185] J. Zhang, W. Ding, and L. He, "Data augmentation and prior knowledge-based regularization for sound event localization and detection", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2019.

[186] P. Pratik, W. J. Jee, S. Nagisetty, R. Mars, and C. Lim, "Sound event localization and detection using CRNN architecture with mixup for model generalization", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2019.

[187] L. Mazzon, M. Yasuda, Y. Koizumi, and N. Harada, "Sound event localization and detection using foa domain spatial augmentation", *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2019.

[188] D. Comminiello, M. Lella, S. Scardapane, and A. Uncini, "Quaternion convolutional neural networks for detection and localization of 3d sound events", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8533–8537.

[189] L. Pi, X. Zheng, C. Zhang, P. Chen, Z. Wang, and X. Li, "U recurrent neural network for polyphonic sound event detection and localization", in *Proceedings of the 5th International Conference on Multimedia Systems and Signal Processing*, 2020, pp. 86–91.

[190] T. N. T. Nguyen, D. L. Jones, and W.-S. Gan, "A sequence match-

ing network for polyphonic sound event localization and detection", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 71–75.

[191] K. Guirguis, C. Schorn, A. Guntoro, S. Abdulatif, and B. Yang, "SELD-TCN: Sound event localization & detection via temporal convolutional networks", *arXiv preprint arXiv:2003.01609*, 2020.

[192] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", in *International Conference on Learning Representations*, 2015.

[193] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[194] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[195] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4694–4702.

[196] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov, "Exploiting image-trained CNN architectures for unconstrained video classification", in *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, September 2015, pp. 60.1–60.13.

[197] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1933–1941.

[198] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2625–2634.

[199] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition", in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4041–4049.

[200] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue, "Modeling spatial-temporal clues in a hybrid deep learning framework for video classification", in *Proceedings of the 23rd ACM International Conference on Multimedia*, 2015, pp. 461–470.

[201] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6299–6308.

[202] C.-Y. Ma, M.-H. Chen, Z. Kira, and G. AlRegib, "TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition", *Signal Processing: Image Communication*, vol. 71, pp. 76–87, 2019.

[203] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. Snoek, "VideoLSTM convolves, attends and flows for action recognition", *Computer Vision and Image Understanding*, vol. 166, pp. 41–50, 2018.

[204] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4489–4497.

[205] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri, "Convnet ar-

chitecture search for spatiotemporal feature learning", *arXiv preprint arXiv:1708.05038*, 2017.

[206] K. Hara, H. Kataoka, and Y. Satoh, "Learning spatio-temporal features with 3D residual networks for action recognition", in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 3154–3160.

[207] M. Zolfaghari, K. Singh, and T. Brox, "Eco: Efficient convolutional network for online video understanding", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 695–712.

[208] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[209] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 305–321.

[210] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos", in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 568–576.

[211] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition", in *European Conference on Computer Vision*. Springer, 2016, pp. 20–36.

[212] L. Fan, W. Huang, C. Gan, S. Ermon, B. Gong, and J. Huang, "End-to-end learning of motion representation for video understanding", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6016–6025.

[213] N. Crasto, P. Weinzaepfel, K. Alahari, and C. Schmid, "MARS: Motion-augmented RGB stream for action recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7882–7891.

[214] L. Wang, W. Li, W. Li, and L. Van Gool, "Appearance-and-relation networks for video classification", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1430–1439.

[215] A. Diba, M. Fayyaz, V. Sharma, M. Mahdi Arzani, R. Yousefzadeh, J. Gall, and L. Van Gool, "Spatio-temporal channel correlation networks for action classification", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 284–299.

[216] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition", in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6202–6211.

[217] Y. Bian, C. Gan, X. Liu, F. Li, X. Long, Y. Li, H. Qi, J. Zhou, S. Wen, and Y. Lin, "Revisiting the effectiveness of off-the-shelf temporal modeling approaches for large-scale video classification", *arXiv preprint arXiv:1708.03805*, 2017.

[218] X. Long, C. Gan, G. De Melo, J. Wu, X. Liu, and S. Wen, "Attention clusters: Purely attention based local feature integration for video classification", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7834–7843.

[219] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen, "Epic-fusion: Audio-visual temporal binding for egocentric action recognition", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 5492–5501.

[220] W. Wang, D. Tran, and M. Feiszli, "What Makes Training Multi-Modal Networks Hard?" in *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition (CVPR), 2020.

[221] S. Pouyanfar, T. Wang, and S.-C. Chen, "A multi-label multimodal deep learning framework for imbalanced data classification", in *IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2019, pp. 199–204.

[222] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue, "Multi-stream multi-class fusion of deep networks for video classification", in *Proceedings of the 24th ACM International Conference on Multimedia*, 2016, pp. 791–800.

[223] Y. Tian, Y. Cao, J. Wu, W. Hu, C. Song, and T. Yang, "Multi-cue combination network for action-based video classification", *IET Computer Vision*, vol. 13, no. 6, pp. 542–548, 2019.

[224] X. Long, C. Gan, G. De Melo, X. Liu, Y. Li, F. Li, and S. Wen, "Multimodal keyless attention fusion for video classification", in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[225] B. Vanderplaetse and S. Dupont, "Improved soccer action spotting using both audio and video streams", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 896–897.

[226] F. Xiao, Y. J. Lee, K. Grauman, J. Malik, and C. Feichtenhofer, "Audiovisual SlowFast networks for video recognition", *arXiv preprint arXiv:2001.08740*, 2020.

[227] Y.-B. Lin, Y.-J. Li, and Y.-C. F. Wang, "Dual-modality seq2seq network for audio-visual event localization", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2002–2006.

[228] Y. Wu, L. Zhu, Y. Yan, and Y. Yang, "Dual attention matching for audio-visual event localization", in *Proceedings of the IEEE Interna-*

tional Conference on Computer Vision (ICCV), 2019, pp. 6292–6300.

[229] J. Ramaswamy and S. Das, "See the Sound, Hear the Pixels", in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2970–2979.

[230] H. Xuan, Z. Zhang, S. Chen, J. Yang, and Y. Yan, "Cross-Modal Attention Network for Temporal Inconsistent Audio-Visual Event Localization", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 279–286.

[231] J. Wang, X. Peng, and Y. Qiao, "Cascade multi-head attention networks for action recognition", *Computer Vision and Image Understanding*, p. 102898, 2020.

[232] J. Li, X. Liu, W. Zhang, M. Zhang, J. Song, and N. Sebe, "Spatio-temporal attention networks for action recognition and detection", *IEEE Transactions on Multimedia*, 2020.

[233] H. Yang, C. Yuan, L. Zhang, Y. Sun, W. Hu, and S. J. Maybank, "STA-CNN: convolutional spatial-temporal attention learning for action recognition", *IEEE Transactions on Image Processing*, vol. 29, pp. 5783–5793, 2020.

[234] S. Liu, X. Ma, H. Wu, and Y. Li, "An end to end framework with adaptive spatio-temporal attention module for human action recognition", *IEEE Access*, vol. 8, pp. 47 220–47 231, 2020.

[235] M. A. Jalal, W. Aftab, R. K. Moore, and L. Mihaylova, "Dual stream spatio-temporal motion fusion with self-attention for action recognition", in *22th International Conference on Information Fusion (FUSION)*. IEEE, 2019, pp. 1–7.

[236] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1501–1510.

[237] T. Kim, I. Song, and Y. Bengio, "Dynamic layer normalization for adaptive neural acoustic modeling in speech recognition", in *Proceedings of Interspeech*, 2017, pp. 2655–2659.

[238] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, "Modulating early visual processing by language", in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 6594–6604.

[239] J. Abdelnour, G. Salvi, and J. Rouat, "From visual to acoustic question answering", *arXiv preprint arXiv:1902.11280*, 2019.

[240] A. Kumar, M. Khadkevich, and C. Fügen, "Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes", in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2018, pp. 326–330.

[241] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer", in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[242] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset", *arXiv preprint arXiv:1705.06950*, 2017.

[243] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne", *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[244] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "CNN architectures for large-scale audio classification", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 131–135.

[245] P. Zhou, W. Yang, W. Chen, Y. Wang, and J. Jia, "Modality attention for end-to-end audio-visual speech recognition", in *IEEE Interna-*

tional Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 6565–6569.

[246] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning", in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[247] J.-B. Delbrouck, N. Tits, M. Brousmiche, and S. Dupont, "A transformer-based joint-encoding for emotion recognition and sentiment analysis", in *Second Grand-Challenge and Workshop on Multimodal Language (Challenge-HML)*. Seattle, USA: Association for Computational Linguistics, Jul. 2020, pp. 1–7.

[248] J. Ren, Y. Hu, Y.-W. Tai, C. Wang, L. Xu, W. Sun, and Q. Yan, "Look, listen and learn—a multimodal LSTM for speaker identification", in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[249] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep multiple instance learning for image classification and auto-annotation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3460–3469.

[250] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras

[251] S. Brodeur, E. Perez, A. Anand, F. Golemo, L. Celotti, F. Strub, J. Rouat, H. Larochelle, and A. Courville, "HoME: A household multimodal environment", in *Visually-Grounded Interaction and Language Workshop (NIPS)*, 2017.

[252] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "AI2-THOR: An Interactive 3D Environment for Visual AI", *arXiv*, 2017.

List of Figures

List of Tables

This thesis was made using a customized version of “hepthesis”.